Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
Στην Ασφάλεια Υπολογιστών και Δικτύων

Cloud Forensics Challenges – Log Format Unification - The CADF Case

Νικόλαος Δαλέζιος

Επιβλέπων Καθηγητής
Σταύρος Σιαηλής

Μάιος 2019

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Cloud Forensics Challenges – Log Format Unification - The CADF Case

Νικόλαος Δαλέζιος

Επιβλέπων Καθηγητής
Σταύρος Σιαηλής

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στην Ασφάλεια Υπολογιστών και Δικτύων

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2019

ΛΕΥΚΗ ΣΕΛΙΔΑ

Περίληψη

Η αύξηση των χρηστών των διαδικτυακών υπηρεσιών και των αναγκών τους σε χώρο και πόρους, έχει οδηγήσει τους παρόχους των υπηρεσιών σε μεταφορά της υποδομής τους στα εικονικά συστήματα και το Cloud. Το χαμηλό κόστος αλλά κυρίως η ανωνυμία που προσφέρει προσελκύει τους κυβερνοεγκληματίες δίνοντας τους ένα νέο πεδίο δράσης.

Η επιστήμη της ψηφιακής δικανικής δεν έχει φτάσει ακόμα τεχνολογικά σε ικανοποιητικό επίπεδο ως προς την εξιχνίαση των ηλεκτρονικών εγκλημάτων στο cloud. Τα πλεονεκτήματα που παρέχει το cloud για τους χρήστες του αποτελούν ταυτόχρονα προβλήματα για τη δικανική. Το ινστιτούτο NIST απαριθμεί 65 ανοιχτές προκλήσεις.

Η παρούσα μεταπτυχιακή διατριβή καταπιάνεται με μία από αυτές τις προκλήσεις. Μελετά την ενοποίηση της μορφής στα αρχεία καταγραφής σε πλατφόρμες cloud. Αρχικά γίνεται μία βιβλιογραφική ανασκόπηση πάνω στα ζητήματα του cloud, των ηλεκτρονικών εγκλημάτων σε αυτό και των προτεινόμενων λύσεων. Στη συνέχεια θεμελιώνονται εννοιολογικά η υποδομή και τα μοντέλα του cloud με ταυτόχρονη αλλά σύντομη επισκόπηση των πιο σημαντικών πλατφόρμων, αλλά και των ελάχιστων εργαλείων για έρευνα ψηφιακών πειστηρίων στο cloud.

Το μοντέλο αποτύπωσης συμβάντων που ερευνάται είναι το CADF στο οποίο πραγματοποιείται ανάλυση με σκοπό την κατανόησή του. Κατά την μελέτη περίπτωσης της πλατφόρμας OpenStack ενεργοποιείται το μοντέλο, έτσι ώστε να μελετηθεί στην πράξη. Με βάση την αποκτηθείσα γνώση υλοποιείται η βασική λειτουργικότητα του CADF στην πλατφόρμα Apache CloudStack. Κατά την υλοποίηση προκύπτουν κάποιες προτάσεις βελτίωσης προς την ίδια την πλατφόρμα, έτσι ώστε να είναι φιλικότερη προς τους ερευνητές τέτοιων συμβάντων. Επιπλέον εκτελούνται σενάρια δοκιμών τα οποία παράγουν αρχεία συμβάντων CADF για περεταίρω μελέτη.

Η παρούσα διατριβή καταλήγει συγκρίνοντας το υλοποιηθέν μοντέλο στο CloudStack με το υπάρχον, αναδεικνύοντας τα ζητήματα που προέκυψαν κατά την ανάπτυξη. Το μοντέλο ελέγχεται σε σχέση με τις οδηγίες ACPO για το χειρισμό των ηλεκτρονικών πειστηρίων. Τέλος παρουσιάζεται ένα αυτοματοποιημένο εργαλείο ανάγνωσης αρχείων καταγραφής, το C.Lo.D., που εντοπίζει αυτόματα CADF καταχωρήσεις τις οποίες και εισάγει σε NoSQL βάση δεδομένων για εκτέλεση ερωτημάτων.

Summary

Increase of internet services users and their need for space and computing resources has led service providers to the expansion of their infrastructure towards virtual systems and cloud. Low cost and mainly anonymity attract cybercriminals, providing them a new field of action.

Digital Forensics Science has not yet achieved a satisfactory technological level regarding cloud digital crime investigation. Most of the advantages of the cloud use – from the user perspective – are at the same time problems for the forensics. NIST Institute numbers 65 open challenges on cloud forensics.

The present thesis deals with one of these challenges. Its field of study is the "Log Unification on Cloud Platforms" issue. At first, a bibliography review is conducted in relation to cloud issues, as well as digital crime on the cloud and recommended solutions. Next, cloud terminology is being explained while at the same time an overview of the most important platforms is conducted alongside with presenting the very few cloud forensics tools.

The proposed model is Cloud Auditing Data Federation (CADF), which is being analyzed, aiming at providing a broad understanding of its use. While using the OpenStack platform, CADF event logging is activated so that the model can be observed in action. Having an in-depth understanding of the model's internals and mappings an effort is made to implement the CADF functionality on the Apache CloudStack platform from scratch. During the implementation process, a number of issues occurred and a few suggestions and improvements are proposed in order for CloudStack to become forensically friendlier. Additionally, a series of test scripts are executed so that a CADF dataset is created for further studying and experimentation.

The present thesis concludes comparing the existing CloudStack event model and the implemented CADF event model, while pointing out the issues and problems that came up during the development process. The proposed model is being examined in correlation with the principles of ACPO on handling digital evidence. Finally, an automated parsing tool/CADF event consumer, named C.Lo.D. is being presented. CLoD parses log files and extracts CADF event records. These records are stored in a NoSQL database so that an investigator can execute queries on the data.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή Δρ. Σταύρο Σιαηλή για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου το συγκεκριμένο θέμα, τη συνεχή καθοδήγηση και την πολύτιμη βοήθειά του. Επίσης θα ήθελα να ευχαριστήσω τη σύζυγό μου Γρηγορία και τα παιδιά μου, Μαρίλια και Μάρκο, για την υπομονή και κατανόηση που έδειξαν καθ' όλη τη διάρκεια του μεταπτυχιακού προγράμματος.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Η συνεχής ανάγκη της μείωσης του κόστους των υπηρεσιών, της ενοποίησης αυτών ανάμεσα σε σταθερές και φορητές συσκευές καθώς και οι συνεχώς αυξανόμενες απαιτήσεις σε αποθηκευτικό χώρο και υπολογιστική ισχύ, έχουν οδηγήσει τις εταιρείες προς την κατεύθυνση των νεφοϋπολογιστικών συστημάτων. Όπως έχει επισημάνει ο Dykstra [1], εκεί όπου συγκεντρώνονται οι άνθρωποι, τα δεδομένα και το χρήμα, εκεί κατευθύνεται και το έγκλημα. Ο όγκος της επεξεργαστικής διαδικασίας μεταφέρεται στο cloud, καθιστώντας επιτακτικό τον καθορισμό διαδικασιών δικανικής ανάλυσης αλλά και επίλυσης των ανοιχτών προκλήσεων/περιορισμών (challenges/limitations) της επιστήμης της δικανικής στο νέο αυτό περιβάλλον.

Στην εποχή των Big Data, του Virtualization και του Cloud Computing ο όγκος της αποθηκευμένης πληροφορίας είναι δυσανάλογα μεγάλος σε σχέση με την ταχύτητα μεταφοράς αυτής. Η διαδικασία της απόκτησης (acquisition) κατά τη διάρκεια της έρευνας για ηλεκτρονικά πειστήρια είναι η πιο καθοριστική για την πορεία αυτής, αφού θα παράσχει στον ερευνητή το αντικείμενο στο οποίο θα πραγματοποιήσει την forensics analysis. Οι περιορισμοί που θέτει το μέγεθος των δεδομένων είναι χρονικοί, κόστους και εγκυρότητας.

Τα αρχεία καταγραφής ενός συστήματος (log files) αποτελούν κρίσιμο συστατικό για την αποσφαλμάτωση και την παρακολούθηση της λειτουργίας και της τρέχουσας κατάστασης του. Κατά τη διάρκεια μίας εγκληματολογικής έρευνας τα αρχεία αυτά πιθανόν να περιέχουν σημαντικές πληροφορίες σχετιζόμενες με το διερευνώμενο συμβάν. Πρόκειται για έναν από τους βασικούς “μάρτυρες” για το τι συνέβαινε στο σύστημα μία συγκεκριμένη χρονική στιγμή.

Αποτελεί γενική παραδοχή πως η διαδικασία ανακάλυψης και συλλογής ψηφιακών πειστηρίων σε περιβάλλοντα cloud διαφέρει από αυτή στα οικιακά/εταιρικά υπολογιστικά συστήματα. Η διαφορά έγκειται στην μη ύπαρξη εργαλείων, διαδικασιών αλλά κυρίως στο γεγονός της πιθανής γεωγραφικής διασποράς του υπό εξέταση συστήματος. Η μη φυσική πρόσβαση σε ένα σύστημα απαιτεί την παροχή υπό συγκεκριμένες συνθήκες, τεχνικές και

νομικές, απομακρυσμένης πρόσβασης. Στο σημείο αυτό εγείρονται ερωτήματα σχετικά με τη διασφάλιση των απαιτήσεων της ακεραιότητας και της αυθεντικότητας των δεδομένων.

Ειδικότερα σε περιβάλλοντα cloud, όπου ο όγκος της πληροφορίας των αρχείων καταγραφής είναι τεράστιος, η χρήσιμη πληροφορία μπορεί να μην εντοπιστεί εύκολα. Επιπρόσθετα τα ζητήματα που απορρέουν από τον ίδιο το σχεδιασμό του cloud, όπως ο κατακερματισμός, η γεωγραφική διασπορά και οι διαφορετικές υλοποιήσεις δυσχεραίνουν τον εντοπισμό της χρήσιμης πληροφορίας.

1.1 Καθορισμός του Προβλήματος

Το 2014, οι Iorga και Simmon συγκέντρωσαν σε μία αναφορά για το ινστιτούτο NIST τις προκλήσεις, τα προβλήματα και τα ανοιχτά ζητήματα – 65 συνολικά - τα οποία αντιμετωπίζουν οι ερευνητές κατά τη διάρκεια της μελέτης και δικανικής ανάλυσης συμβάντων που έλαβαν χώρα στο cloud [2].

Η πρόκληση με αριθμό 6 (έξι) έχει τίτλο “Log Format Unification”. Στα περιβάλλοντα νεφοϋπολογισμού είναι δύσκολη η ύπαρξη ενός γενικού προτύπου των αρχείων καταγραφής. Η υλοποίηση του μηχανισμού και του τρόπου οργάνωσης των αρχείων καταγραφής είναι αποκλειστική αρμοδιότητα του παρόχου. Ο κάθε πάροχος έχει διαφορετική αντίληψη περί βέλτιστης μορφής και μηχανισμού των αρχείων καταγραφής. Η ενοποίηση του μηχανισμού και της μορφής καταγραφής είναι τόσο δύσκολη ώστε ακόμα και η μετατροπή υπαρχόντων αρχείων καταγραφής από τη μορφή ενός παρόχου στη μορφή ενός άλλου είναι αδύνατη.

Ο οργανισμός Distributed Management Task Force (DMTF) απάντησε δημιουργώντας και προτείνοντας το Cloud Auditing Data Federation (CADF) format ως ένα standard για τη μορφή των αρχείων καταγραφής [3]. Το ίδρυμα OpenStack (https://www.openstack.org/) υιοθέτησε στο ομώνυμο έργο (OpenStack) το πρότυπο CADF, δημιουργώντας τη βιβλιοθήκη PyCADF (https://github.com/openstack/pycadf), ενώ μία νέα έρευνα [4] (manuscript) ενσωματώνει το πρότυπο στο έργο CloudStack (https://cloudstack.apache.org/) του ιδρύματος Apache (http://apache.org/).

Η παρούσα μεταπτυχιακή διατριβή εξετάζει

1. Αν μπορεί το πρότυπο CADF να υλοποιηθεί από τρίτους παρόχους υπηρεσιών cloud όπως το Apache CloudStack ικανοποιώντας την απαίτηση της ανοιχτότητας (openness)

2. Με ποιο τρόπο μπορεί να εξασφαλιστεί η διαφάνεια και η καθολικότητα στην υλοποίηση
3. Τι μετρικές, δοκιμές και use cases απαιτούνται για την υποστήριξη της πλήρους αποδοχής του CADF

1.2 Δομή της Μεταπτυχιακής Διατριβής

Η παρούσα διπλωματική διατριβή δομείται σε κεφάλαια. Στο Κεφάλαιο 2 γίνεται επισκόπηση της υπάρχουσας βιβλιογραφίας σχετικά με το cloud και την επιστήμη της δικανικής σε αυτό. Στο Κεφάλαιο 3 πραγματοποιείται η εννοιολογική θεμελίωση όλων των απαραίτητων όρων που θα χρησιμοποιηθούν στη διατριβή ενώ στο Κεφάλαιο 4 γίνεται μία σύντομη αναφορά στις πιο γνωστές Cloud Platforms και Virtualization Platforms. Στο Κεφάλαιο 5 απαριθμούνται τα εργαλεία Cloud Forensics, οι βασικές τους δυνατότητες αλλά και οι περιορισμοί τους. Στο Κεφάλαιο 6, στο πλαίσιο της ανάλυσης του Cloud Auditing γίνεται ανάλυση του μοντέλου CADF που είναι και το κύριο θέμα της διατριβής. Στη συνέχεια, στο Κεφάλαιο 7 απεικονίζεται η τρέχουσα τεχνολογία στην αγορά ως προς το μοντέλο IaaS . Στο Κεφάλαιο 8 δοκιμάζεται η πλατφόρμα OpenStack με το μοντέλο CADF. Στο Κεφάλαιο 9 πραγματοποιείται ανάλυση και μελέτη της πλατφόρμας Apache CloudStack. Υλοποιείται το μοντέλο CADF, καταγράφονται οι παρατηρήσεις και τα αποτελέσματα και η συσχέτιση με τις αρχές ACPO. Τέλος, στο Κεφάλαιο 10 παρουσιάζονται τα συμπεράσματα τις διατριβής και δίνονται κατευθύνσεις για μελλοντική δουλειά.

Κεφάλαιο 2
Βιβλιογραφία

Η πρώτη επίσημη και ολοκληρωμένη προσπάθεια συγκέντρωσης των ανοιχτών ζητημάτων στα cloud forensics πραγματοποιήθηκε για λογαριασμό του NIST, το 2014 [2]. Μία νέα μελέτη, το 2015 επισήμανε την εξάρτηση της διαθεσιμότητας και του είδους των αρχείων καταγραφής από τον πάροχο υπηρεσιών Cloud [5]. Ειδικότερα στα μοντέλα SaaS και PaaS, είναι αποκλειστικά στη δικαιοδοσία του παρόχου η διάθεση των αρχείων καταγραφής, χωρίς φυσικά να υπάρχει καμία διαβεβαίωση για την ακεραιότητα των παραδοτέων και για το αν μπορούν να γίνουν αποδεκτά από τις δικαστικές αρχές..

Το 2011, μελετώντας την cloud πλατφόρμα Eucalyptus, καταγράφηκε σε logs η αλληλεπίδραση των συστατικών της. Με αυτόν τον τρόπο εντοπίστηκε επίθεση DDoS που ξεκίνησε μέσα από το εξεταζόμενο cloud [6]. Την ίδια χρονιά, προτάθηκε για πρώτη φορά από τους Birk and Wegener, η χρήση read-only API, με στόχο ο πελάτης του cloud να έχει τη δυνατότητα λήψης δεδομένων από αυτό και να τα παράσχει ο ίδιος στους ερευνητές [7].

Σύμφωνα με τους Marty και Zawoad, λόγω της ετερογενούς φύσης των αρχείων καταγραφής στο cloud είναι ανάγκη να καθορίζονται τουλάχιστον τα εξής [8][9] :

- Πότε γίνεται καταγραφή συμβάντος
- Τι ακριβώς καταγράφεται
- Ποια μορφή έχει η εγγραφή

Επίσης ορίζονται ως ελάχιστα πεδία της εγγραφής τα παρακάτω

- Χρονοσήμανση / timestamp
- Εφαρμογή / application
- Χρήστης / user
- Αναγνωριστικό συνόδου / session id
- Σοβαρότητα / severity
- Αιτία / reason
- Κατηγοριοποίηση / categorization

Η σύνταξη της εγγραφής συνίσταται να ακολουθεί τη μορφή “key-value”. Με αυτόν τον τρόπο απαντώνται τα 4 από τα 7 W’s of audit (What, When, Who, Why).

Σε μελέτη αναφορικά με τη σχέση των ACPO οδηγιών και του cloud [10], παρουσιάζονται οι διαφοροποιήσεις και τα κενά που προκύπτουν κατά την προσπάθεια εφαρμογής των 4 αρχών της ACPO στην έρευνα, όταν αυτή αφορά περιβάλλοντα cloud. Η ερευνητική μεθοδολογία και τα βήματα διαφοροποιούνται ανάλογα με το deployment model (private, public, hybrid, community). Ενδεικτικά αναφέρεται πως σε private cloud οι τοποθεσίες φύλαξης των δεδομένων είναι γνωστές και προσβάσιμες με την κατάλληλη άδεια. Επίσης το προσωπικό των οργανισμών που διαχειρίζονται το cloud γνωρίζει τις διαδικασίες καταγραφής. Αντίθετα στα public clouds ο οργανισμός που διαχειρίζεται το cloud το διαθέτει για χρήση στους πελάτες του. Οι πελάτες δε γνωρίζουν λεπτομέρειες για τον εσωτερικό τρόπο λειτουργίας του οργανισμού και του ίδιου του cloud.

Όπως έχει ήδη επισημανθεί [11], στο μοντέλο IaaS η ευθύνη της έρευνας αφορά τον πελάτη εκτός από ειδικές περιπτώσεις. Βέβαια σε αυτό ο συντάκτης της παρούσας διατριβής διαφωνεί διότι πρέπει πάντοτε ο CSP να συμμετέχει ενεργά στην έρευνα εκτός από την απλή παροχή βοήθειας. Ο CSP, έχει μεγαλύτερο συμφέρον να διεξαχθεί με επιτυχία η έρευνα από ότι ο πελάτης διότι το ερευνώμενο συμβάν είναι πιθανό να ξεκίνησε από αδυναμία της ίδιας της πλατφόρμας του CSP ή ακόμα χειρότερα να δόθηκε πρόσβαση στα στοιχεία των υπολοίπων πελατών του CSP. Πιθανές επιπτώσεις τέτοιων συμβάντων μπορεί να είναι το πλήγμα στην επιχειρησιακή συνέχεια του CSP ή στην επιχειρηματική δραστηριότητα και φήμη.

Μία διαφορετική προσέγγιση επιχειρήθηκε από το 2015, προτείνοντας ένα ανοιχτό μοντέλο forensics (OCF – Open Cloud Forensics) [9]. Στο μοντέλο OCF συμμετέχουν 4 οντότητες – ο χρήστης, ο CSP, ο ερευνητής και οι δικαστικές αρχές. Με βάση αυτή τη θεώρηση, κάθε πρόσβαση στο cloud παράγει ESI (Electronically Stored Information), μη αποδεκτό από τις δικαστικές αρχές. Για να γίνει αποδεκτό πρέπει ο CSP με συγκεκριμένη διαδικασία να το καταστήσει έγκυρο προς τις δικαστικές αρχές. Η διαδικασία αυτή είναι συνεχής (Continuous Forensics Process). Παρόλο που δεν αναφέρεται ρητά στα αρχεία καταγραφής, η κεντρική ιδέα τα περιλαμβάνει, εφόσον ανήκουν στα δεδομένα που πρέπει να καταστήσει έγκυρα ο CSP.

Οι Pătraşcu και Valeriu Patriciu, παρουσίασαν το 2015 ένα forensics framework που μπορεί να ενσωματωθεί σε υπάρχουσες υποδομές cloud αλλά και σε νέες. Αφού σύγκριναν 2 μορφές απεικόνισης των δεδομένων των συμβάντων - τη “Management Metalanguage” από UnixWare (http://uw714doc.sco.com/en/UDI_spec/m_mgmt.html) και το Common Event Expression (CEE) Language (http://cee.mitre.org/language/1.0-beta1/cls.html) κατέληξαν σε ένα συνδυασμό αυτών. Με αυτόν τον τρόπο αξιοποίησαν τη λειτουργικότητα του Management Language με τη βοήθεια της JSON απεικόνισης των δεδομένων [12]. Στη συνέχεια, χώρισαν την cloud αρχιτεκτονική σε 2 μέρη

- Διαχείριση/Management
- Virtualization

Στο επίπεδο του management πρόσθεσαν ένα Cloud Forensics Module που αναλαμβάνει την καταγραφή των δεδομένων της αλληλεπίδρασης των συστατικών του cloud. Στο επίπεδο του virtualization, εισηγήθηκαν την χρήση Cloud Forensics Interface για τη συλλογή δεδομένων από το εσωτερικό των εικονικών μηχανών.

Αυτή η πρόταση - τουλάχιστον ως προς το μέρος του management - θυμίζει έντονα τη λογική ενός συστατικού τηλεμετρίας. Η πρότασή τους υλοποιήθηκε μόνο για KVM hypervisors. Τα δεδομένα που δύναται να συλλεχθούν υπερβαίνουν κατά πολύ τα log files, αφού παρέχεται δυνατότητα λήψης virtual disks και memory dumps.

Οι Kumar Raju και Geethakumari μελέτησαν το 2016, τη δυνατότητα της συσχέτισης της πληροφορίας που εμπεριέχεται στα αρχεία καταγραφής από τη σκοπιά του ερευνητή δικανικής στην πλατφόρμα OpenStack [13]. Στα αρχεία καταγραφής εφαρμόστηκε διαδικασία κανονικοποίησης με σκοπό την προσωρινή επίλυση του προβλήματος της κοινής και ενιαίας μορφής των αρχείων καταγραφής. Η συσχέτιση της πληροφορίας έγινε σε αρχεία καταγραφής είτε από ομογενή artifacts [1] είτε από ετερογενή.

Οι Sekhar and Murali, το επόμενο έτος, έθιξαν εκ νέου το ζήτημα της εμπιστοσύνης προς τον CSP υλοποιώντας ένα σύστημα Secure Logging Services. Βασιζόμενοι στην ομομορφική κρυπτογραφία για κρυπτογράφηση των εγγραφών των αρχείων καταγραφής τα προστατεύουν από πιθανή αλλοίωσή τους [14]. Αυτή μπορεί να προέλθει είτε από τον ίδιο το CSP είτε από επιτιθέμενο που έχει αποκτήσει πρόσβαση στον CSP και το εσωτερικό του cloud.

Το ίδιο έτος, σε μελέτη προτάθηκε μία προσέγγιση ασφαλούς logging βασιζόμενη σε συλλογή και απεικόνιση των δεδομένων με συνδυασμό 3 αλγορίθμων [15]

- SystemInit - εκτελείται στον CSP και παράγει τα απαραίτητα συστατικά για την κρυπτογράφηση
- KeyGen - εκτελείται από το χρήστη και παράγει ένα είδος ψηφιακής υπογραφής
- SecLogging - Συνδυάζει τα αποτελέσματα των 2 παραπάνω αλγορίθμων και κρυπτογραφεί log blocks αντί για log entries.

Το 2017 αναγνωρίστηκε η ανάγκη εφαρμογής ενός ενιαίου μηχανισμού συλλογής των αρχείων καταγραφής καθώς οι διάφορες λύσεις είτε δεν καλύπτουν τα διαφορετικά

[1] Εφόσον πρόκειται για Virtual Machines τα 3 κύρια artifacts είναι τα VRAM, Service logs και Vdisk.

μοντέλα cloud (IaaS, PaaS, SaaS) είτε δεν ξεπερνούν τα εμπόδια του εντοπισμού, της επισκόπησης και της συσχέτισης της πληροφορίας σε αυτά [16]. Ως η πιο ολοκληρωμένη λύση θεωρήθηκε η προσέγγιση του CADF, ωστόσο αναδείχθηκαν δύο ζητήματα. Το πρώτο αφορά στην απουσία use cases και δοκιμών για να καταδειχθεί αν τα δεδομένα που συλλέγονται με βάση το CADF είναι αρκετά για τη διεξαγωγή έρευνας. Το δεύτερο αφορά στην ύπαρξη υπερβολικά πολλών δεδομένων στα αρχεία καταγραφής, γεγονός που επίσης δυσχεραίνει την έρευνα.

Το πρόβλημα της ενοποίησης των αρχείων καταγραφής αναδείχθηκε εκ νέου στην μελέτη περίπτωσης 3 διαφορετικών παρόχων υπηρεσιών αποθηκευτικού χώρου (Cloud Storage Service Provider), των Amazon Web Services Simple Storage Service (AWS S3), Google Cloud Platform Storage και Microsoft Azure Storage [17]. Η λύση που προτάθηκε είναι η χρήση μίας ενιαίας μορφής/δομής στα αρχεία καταγραφής. Αυτή προκύπτει ως αποτέλεσμα 3 βημάτων

a. Συλλογή αρχείων καταγραφής από τους διαφορετικούς παρόχους
b. Έλεγχος για διπλότυπα στα αρχεία καταγραφής
c. Μετατροπή και κανονικοποίηση στην επιθυμητή μορφή

Η λογική που ακολουθήθηκε παραπέμπει σε αυτή που ακολουθεί το πρότυπο CADF.

Το 2018, παρουσιάστηκε μία εργασία πάνω στην λεπτομερή εξέταση των μηχανισμών καταγραφής στο cloud με σκοπό τη διευκόλυνση της δικανικής έρευνας [4]. Παράλληλα, υλοποιήθηκε το πρότυπο CADF στο σύστημα παροχής cloud υπηρεσιών CloudStack. Στη συνέχεια έγιναν δοκιμές παρείσδυσης (penetration tests) προκειμένου να επαληθευτεί η ορθή λειτουργία της υλοποίησής.

Κεφάλαιο 3
Εννοιολογική θεμελίωση

Το 2011, έπειτα από 15 πρόχειρες εκδόσεις οι Mell και Grance δημοσίευσαν εκ μέρους του National Institute of Standards and Technology (NIST) το έγγραφο NIST Special Publication 800-145 με τίτλο “The NIST Definition of Cloud Computing”. Σε αυτό ορίζεται η έννοια, τα χαρακτηριστικά, τα μοντέλα παροχής και τα μοντέλα ανάπτυξης των υπηρεσιών των συστημάτων νεφοϋπολογισμού (εφεξής “cloud computing”)[18].

3.1 Cloud computing

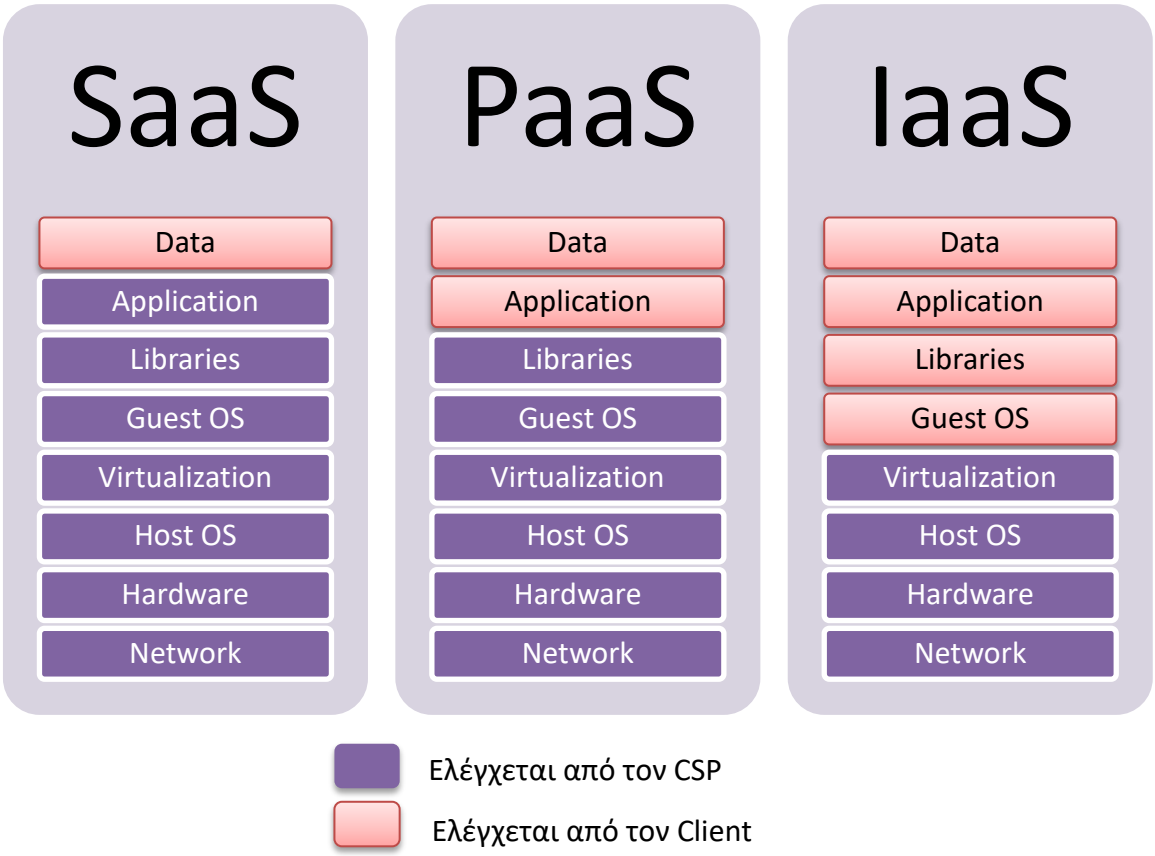
Οι υπηρεσίες cloud computing ορίζονται ως η κατ’ απαίτηση (on-demand) παροχή πλήρως παραμετροποιήσιμων και διαχειρίσιμων υπολογιστικών πόρων – από εφαρμογές μέχρι υποδομή datacenters - κάνοντας χρήση δικτυακής υποδομής. Οι πόροι προσφέρονται στους αιτούντες απαιτώντας την ελάχιστη δυνατή συμμετοχή από τον πάροχο των υπηρεσιών.

3.2 Βασικά χαρακτηριστικά

- Αυτοεξυπηρέτηση κατ’ απαίτηση (On-demand self-service) – η παροχή των πόρων γίνεται χωρίς να απαιτείται ανθρώπινη παρέμβαση από τον πάροχο των υπηρεσιών.
- Ευρεία δικτυακή πρόσβαση (Broad network access) – οι πόροι και οι υπηρεσίες είναι διαθέσιμοι μέσω δικτύου και η πρόσβαση είναι ανεξάρτητη από την πλατφόρμα
- Συγκέντρωση πόρων (Resource pooling) – ο πάροχος των υπηρεσιών cloud προσφέρει τους πόρους (φυσικούς και εικονικούς) με δυναμικό τρόπο, ανάλογα με τις τρέχουσες απαιτήσεις των πελατών. Οι πελάτες δεν έχουν κανέναν έλεγχο και γνώση σχετικά με την τοποθεσία και τη φύση (φυσικός/εικονικός) του παρεχόμενου πόρου
- Άμεση ελαστικότητα (Rapid elasticity) – το μέγεθος και οι δυνατότητες των παρεχόμενων πόρων αυξομειώνονται ανάλογα με τις απαιτήσεις του πελάτη σε ελάχιστο χρόνο
- Μετρήσιμες υπηρεσίες (Measured service) – οι πάροχοι των υπηρεσιών cloud καθορίζουν κάποιες μονάδες μέτρησης ανάλογες με τον τύπο της υπηρεσίας (αποθήκευση, επεξεργαστική ισχύς, δικτυακή κίνηση κλπ)

3.3 Μοντέλα παροχής υπηρεσίας (Service models)

- Software-as-a-Service (Saas) Ο πελάτης της υπηρεσίας κάνει χρήση των εφαρμογών που εκτελούνται στην υπάρχουσα υποδομή του παρόχου. Ο πελάτης δεν έχει πρόσβαση στο δίκτυο, τους διακομιστές, το λειτουργικό σύστημα, τις εγκατεστημένες εφαρμογές και τις ρυθμίσεις των συστατικών της υποδομής. Παράδειγμα SaaS αποτελεί το Microsoft Office 365 (https://products.office.com/en-us/business/small-business-solutions)
- Platform-as-a-Service (Paas) – Ο πάροχος προσφέρει την πλατφόρμα που απαιτείται για την εκτέλεση εφαρμογών παρέχοντας την υποστήριξη των αναγκαίων γλωσσών προγραμματισμού, των βιβλιοθηκών και των εργαλείων. Ο πελάτης δεν έχει πρόσβαση στο δίκτυο, τους διακομιστές και το λειτουργικό σύστημα, αλλά έχει έλεγχο πάνω στις εγκατεστημένες εφαρμογές και τις ρυθμίσεις τους. Παράδειγμα PaaS αποτελεί το Google App Engine (https://cloud.google.com/appengine/)
- Infrastructure-as-a-Service (IaaS) – Οι πόροι όπως επεξεργαστική ισχύς, δίκτυα, λειτουργικά συστήματα και εφαρμογές είναι πλήρως προσβάσιμα από τον πελάτη. Αυτός, έχει στη διάθεσή του μία εικονική υποδομή χωρίς όμως να του δίνεται πρόσβαση στην πραγματική υποδομή του παρόχου. Παράδειγμα IaaS αποτελεί το OpenStack (https://www.openstack.org/)



Σχήμα 1 - Μοντέλα παροχής υπηρεσιών cloud και επίπεδα πρόσβασης

3.4 Μοντέλα ανάπτυξης (Deployment models)

- Ιδιωτικό cloud (Private) – Η υποδομή αποτελεί ιδιοκτησία ενός φορέα/οργανισμού/εταιρείας ο οποίος την διαχειρίζεται όπως επιθυμεί
- Κοινοτικό cloud (Community) – Η υποδομή αποτελεί ιδιοκτησία και διατίθεται αποκλειστικά για τις ανάγκες κάποιας κοινότητας[2]
- Δημόσιο cloud (Public) – Η υποδομή και οι υπηρεσίες είναι διαθέσιμες στο ευρύ κοινό
- Υβριδικό cloud (Hybrid) – Αποτελεί συνδυασμό των παραπάνω εφόσον πρόκειται για υποδομή cloud που αποτελείται από διακριτά συστατικά με το κάθε ένα να ακολουθεί διαφορετικό μοντέλο ανάπτυξης.

3.5 Αναγκαιότητα & χρησιμότητα

Κατά το Μάιο και τον Απρίλιο του 2017, η εταιρεία Gartner (https://www.gartner.com) διεξήγαγε έρευνα σε 699 μικρές και μεσαίες επιχειρήσεις με έδρα τις Η.Π.Α, σχετικά με τη χρήση του cloud. Σύμφωνα με αυτή, το 62% των επιχειρήσεων ήδη χρησιμοποιεί υπηρεσίες cloud και το 33% πρόκειται να υιοθετήσει την τεχνολογία cloud στο άμεσο μέλλον [19]. Μερικοί λόγοι που οδηγούν τις επιχειρήσεις στη χρήση αυτών των υπηρεσιών είναι :

- Η διαθεσιμότητα (uptime and failover) και η επεκτασιμότητα των υποδομών (υπολογιστική ισχύς, αποθηκευτικός χώρος, δικτυακό εύρος ζώνης) ανάλογα με τη χρήση με ελάχιστο κόστος
- Η χρήση συνεργατικών εργαλείων γραφείου (Google G Suite, Microsoft Office 365) ανεξαρτήτως της τοποθεσίας
- Η μη εξάρτηση της εργασίας από το χώρο του γραφείου αφού οι εργασίες μπορούν με τα κατάλληλα εργαλεία να γίνουν από οποιοδήποτε σημείο στον κόσμο
- Η ενημέρωση και εγκατάσταση λειτουργικών συστημάτων, λογισμικού εφαρμογών και η διαχείριση των αναβαθμίσεων είναι κατά περίπτωση στην ευθύνη του παρόχου
- Η πρόσβαση σε στατιστικά, δεδομένα και αποτελέσματα επιχειρηματικής έρευνας (Big Data) και σε εργαλεία ανάγνωσης και ανάλυσής τους
- Η παροχή αποθήκευσης αντιγράφων ασφαλείας και επαναφοράς των αντιγράφων σε περίπτωση καταστροφής και η υποστήριξη των σχεδίων επιχειρησιακής συνέχειας (Business Continuity plan, Disaster Recovery as a Service)

[2] Με την έννοια της ομάδας ατόμων με κοινό σκοπό

Σύμφωνα με τη δημοσκόπηση “Hype Cycle for Emerging Technologies, 2018” της εταιρείας Gartner και τις 5 τάσεις που προέκυψαν από αυτή, η χρήση του cloud θα επεκταθεί για να χρησιμοποιηθεί ως πλατφόρμα τεχνητής νοημοσύνης [20]. Η ίδια εταιρεία (Gartner) στην έρευνα “Moving to a Software Subscription Model” [21] υποστηρίζει πως το μοντέλο αγοράς, αδειοδότησης και αναβάθμισης του λογισμικού θα αντικατασταθεί από το μοντέλο online συνδρομής λογισμικού. Μάλιστα θεωρεί πως μέχρι το 2020 το 80% των εταιρειών παραγωγής λογισμικού θα υιοθετήσει το νέο μοντέλο. Φυσικά αυτό δε μπορεί να γίνει χωρίς τη χρήση του cloud και συγκεκριμένα του μοντέλου SaaS.

3.6 Συσχετισμός με το ηλεκτρονικό έγκλημα

Το ηλεκτρονικό έγκλημα και το Διαδίκτυο είναι έννοιες αλληλένδετες. Ειδικότερα, το Διαδίκτυο ανάλογα με τον τρόπο που χρησιμοποιείται από τους κακόβουλους χρήστες διαχωρίζει το ηλεκτρονικό έγκλημα σε

- Cyber-assisted – πχ Social Engineering password
- Cyber-enabled – πχ Μαζικές ηλεκτρονικές απάτες
- Cyber-dependent – πχ DDoS επιθέσεις

Τα ίδια οφέλη που παρέχει η χρήση του Cloud στις επιχειρήσεις μπορούν να χρησιμοποιηθούν από τους ηλεκτρονικούς εγκληματίες. Η αυξημένη υπολογιστική ισχύς, η αυξημένη χωρητικότητα, η εξοικονόμηση ενέργειας και κόστους, η ανωνυμία και η ευελιξία είναι μερικά μόνο από τα χαρακτηριστικά που μπορούν να εκμεταλλευτούν κακόβουλοι χρήστες διευρύνοντας έτσι τις δυνατότητες τους και τους στόχους τους. Η δημιουργία και η χρήση botnets, data-mining, crypto-mining και CnC centers είναι πλέον ευκολότερη από ποτέ. Εκτός όμως από τα εργαλεία των εγκληματιών αυξάνεται σημαντικά και το εύρος των στόχων. Από τη στιγμή που ένας πελάτης θα αποφασίσει να χρησιμοποιήσει τις υπηρεσίες cloud, πρέπει να αποδεχτεί πως τα δεδομένα του δεν βρίσκονται πλέον αποκλειστικά και μόνο στη δική του διάθεση και δικαιοδοσία, ούτε μόνο σε μία συγκεκριμένη τοποθεσία. Όταν δε, στην παραπάνω «εξίσωση» προστεθεί και ο παράγοντας IoT/IoA, όπου κάθε συσκευή μπορεί να συνδέεται και να επικοινωνεί με δίκτυο – πολλές φορές χρησιμοποιώντας την υποδομή cloud – τότε οι στόχοι (σε αριθμό) αλλά και η πολυπλοκότητα των επιθέσεων εκτινάσσονται [22].

Σημαντικό κίνητρο για τη χρήση του cloud στο ηλεκτρονικό έγκλημα αποτελεί επίσης η έλλειψη εργαλείων και διαδικασιών εγκληματολογικής ανάλυσης σε συνδυασμό με τα ανοιχτά ζητήματα [2], [23]. Πολλές φορές δε χρειάζεται καν οι εγκληματίες να λάβουν αντίμετρα (antiforensics), αρκεί να χρησιμοποιήσουν ως βάση για την οποιαδήποτε κακόβουλη ενέργεια υποδομές που βρίσκονται εκτός δικαστικής και διοικητικής δικαιοδοσίας των αρχών.

3.7 Αρχεία Καταγραφής

Η ανάγκη για καταγραφή συμβάντων είναι επιτακτική σε όλων των ειδών τα έργα, είτε πρόκειται για απλά scripts μερικών δεκάδων γραμμών κώδικα, είτε πρόκειται για έργα ευρείας κλίμακας. Σημαντικότερο όμως ζήτημα αποτελεί το ΤΙ (What) θα καταγραφεί, ΠΩΣ (How) θα αποτυπωθεί και σε τι μορφή (format), ΠΟΥ (Where) και με ποιο τρόπο θα αποθηκευτεί. Κατά καιρούς έχουν προταθεί και χρησιμοποιηθεί από οργανισμούς και εταιρείες διάφορα formats αρχείων καταγραφής. Ο Gagliardi Rocco σε άρθρο του σε εταιρικό ιστολόγιο συγκεντρώνει μερικά από τα σημαντικότερα Logging formats [24].

Format	Type	Proposed by	Year	Status
CBE	Proprietary	IBM, Cisco	2003	Dead
CIM	Open	DMTF	2005	Alive
CEF	Proprietary	ArcSight	2006	Alive
CEE	Open	MITRE	2007	Killed
OLF	Proprietary	eIQNetworks	2007	Dead
WELF	Proprietary	WebTrends	2008	Alive
LEEF	Proprietary	Q1 Labs	2013	Alive
CADF	Open	DMTF	2015	Alive

Σχήμα 2 - Συνοπτικός πίνακας Logging formats

Είναι ενδεικτικό ότι από το 2003 κι έπειτα στα ανοιχτά πρότυπα logging ξεχωρίζουν τα CIM και CADF. Και τα δύο είναι έργα του DMTF αλλά οι ομοιότητες σταματούν εκεί. Το CIM είναι ένα πρότυπο περιγραφής χαρακτηριστικών λογισμικού και υλικού με τρόπο ενιαίο για διαφορετικούς κατασκευαστές. Τα προϊόντα διαφορετικών κατασκευαστών μπορούν να περιγράφονται χρησιμοποιώντας κοινά πεδία όπως device name, serial number, model κλπ. [25]. Το CADF μελετάται ενδελεχώς στη συνέχεια της μεταπτυχιακής διατριβής.

Σχήμα 3 - Πόσο σημαντικά είναι τα αρχεία καταγραφής;

3.7.1 Syslog

Μέχρι και σήμερα (2019), το πιο μαζικά αποδεκτό πρότυπο Logging είναι το syslog, όπως αυτό περιγράφηκε στο RFC 5424. Υλοποιείται με μικρές διαφοροποιήσεις από διάφορα λειτουργικά συστήματα. Παρόλο που αναπτύχθηκε το 1983 ως μέρος του *sendmail* project (http://www.sendmail.org/) επίσημα καθιερώθηκε ως πρότυπο το Μάρτιο του 2009 [26]. Η πλειοψηφία των εργασιών που εκτελούνται από τα σύγχρονα λειτουργικά συστήματα και τις εφαρμογές τους χρησιμοποιούν για τα αρχεία καταγραφής τους τη μορφή του syslog. Βασικά πεδία του syslog μηνύματος είναι τα facility code[3], το severity level[4], το ID της διεργασίας, η χρονοσήμανση, το hostname και η IP διεύθυνση του πόρου που καταγράφει το συμβάν.

[3] Κωδικός που καθορίζει τον τύπο της εφαρμογής που κάνει την καταγραφή (kernel message, user message, system daemon κλπ.)

[4] Κωδικός που καθορίζει τη σοβαρότητα του συμβάντος (alert, error, critical, information κλπ.)

Κεφάλαιο 4
Cloud Layers

Ο όρος cloud είθισται να χρησιμοποιείται λανθασμένα και γενικευμένα, άλλοτε εννοώντας την ίδια την εικονική υποδομή, άλλοτε το λογισμικό που δημιουργεί την υποδομή και άλλοτε το λογισμικό διαχείρισης. Η ανάγκη καθορισμού συγκεκριμένων ορισμών, ερμηνειών και μεθοδολογιών καλύπτεται από ομάδες εργασίας που κατά καιρούς δημοσιεύουν αναφορές - οδηγούς.

4.1 OMG Cloud Working Group

Το OMG (Object Management Group) Cloud Working Group (https://www.omg.org/) – πρώην Cloud Standards Customer Council (CSCC) - είναι ένας μη κερδοσκοπικός οργανισμός που έχει σκοπό την παροχή καθοδήγησης σε ζητήματα που άπτονται της υιοθέτησης της τεχνολογίας cloud, σε αρχιτεκτονικές cloud, σε πρότυπα και σε καλές πρακτικές με σκοπό τη διαλειτουργικότητα ανεξάρτητα τον πάροχο των υπηρεσιών. Από το 2011 έως και σήμερα έχει δημοσιεύσει 28 μελέτες/εργασίες χωρισμένες στις εξής κατηγορίες:

- Πρακτικοί οδηγοί (Practical guides)
- Διαλειτουργικότητα και μετάπτωση (Cloud interoperability and migration)
- Σύμφωνα παροχής υπηρεσιών cloud (Cloud service agreements)
- Ασφάλεια cloud (Cloud security)
- Σχετικές με cloud τεχνολογίες (Industry and related cloud technologies)

4.2 Cloud Platforms

Ως πλατφόρμα υπηρεσιών cloud ορίζεται το σύνολο που περιλαμβάνει λειτουργικό σύστημα, το λογισμικό hypervisor, το λογισμικό εφαρμογών και λογισμικό διαχείρισης υπηρεσιών και χρηστών του cloud. Το λογισμικό αυτό μαζί το υλικό – συνήθως κάποιον server – “πάνω” στον οποίο εκτελείται ονομάζεται “cloud stack”. Είναι σύνηθες να συγχέονται οι έννοιες Cloud Platform και Cloud Management Platform. Το Cloud

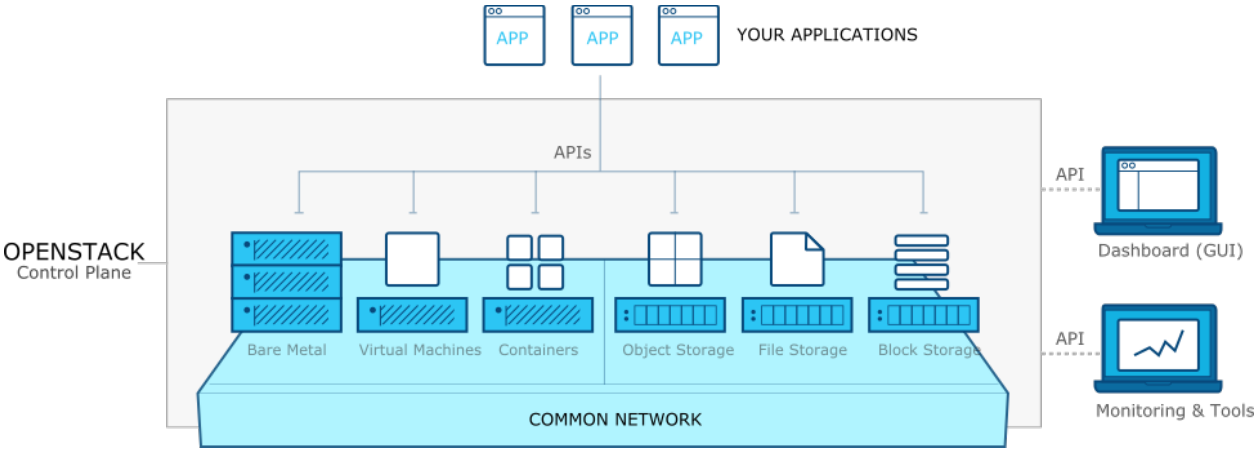
Management Platform είναι υποσύνολο, και βασικό συστατικό του Cloud Platform. Μερικές από τις πιο διαδεδομένες [5]cloud platforms είναι οι

- OpenStack - https://www.openstack.org/
- Apache CloudStack - https://cloudstack.apache.org/
- Amazon Web Services - https://aws.amazon.com/
- Microsoft Azure - https://azure.microsoft.com/en-us/
- Google Cloud Platform - https://cloud.google.com/
- OpenNebula - https://opennebula.org/

Ιδιαίτερο ενδιαφέρον παρουσιάζει ο τρόπος που οι παραπάνω πλατφόρμες αυτοπροσδιορίζονται στους ιστότοπους τους.

4.2.1 OpenStack

Λογισμικό δημιουργίας και διαχείρισης cloud για έλεγχο υπολογιστικής ισχύος, αποθηκευτικού χώρου και δικτυακών πόρων – κυρίως σε datacenter. Η διαχείριση γίνεται μέσω Πίνακα Ελέγχου (dashboard).



Σχήμα 4 – Βασική δομή OpenStack

4.2.2 Apache CloudStack

Το Apache CloudStack είναι λογισμικό ανοιχτού κώδικα σχεδιασμένο για *υλοποίηση και διαχείριση δικτύων εικονικών μηχανών* βασισμένα στο μοντέλο IaaS. Περιλαμβάνει ολόκληρη τη στοίβα των απαραίτητων συστατικών, όπως έλεγχο υπολογιστικής ισχύος, έλεγχο δικτυακών πόρων κλπ. Επίσης παρέχει web interface και γραμμή εντολών για τη διαχείριση του cloud από τους χρήστες.

4.2.3 Amazon Web Services

Ασφαλής *πλατφόρμα υπηρεσιών cloud* που παρέχει υπολογιστική ισχύ, αποθηκευτικό χώρο, βάσεις δεδομένων και εργαλεία απαραίτητα για τη λειτουργία των επιχειρήσεων. Η

[5] Η σειρά και η επιλογή είναι τυχαία και δεν αποτελούν κατάταξη ή ταξινόμηση

Amazon αναφέρει πως πρόκειται για πλατφόρμα που μπορεί να καλύψει κάθε ανάγκη και περίπτωση (use case). Η διαχείριση γίνεται μέσω ειδικών εργαλείων για κάθε περίπτωση/πόρο και ανήκουν στην κατηγορία προϊόντων AWS Management Tools.

4.2.4 Microsoft Azure

Ευέλικτο λογισμικό, ικανό να διαχειριστεί σε μεγάλη εταιρική κλίμακα τις απαιτήσεις μίας *cloud computing πλατφόρμας*. Είναι συνεχώς επεκτάσιμο, καθώς το σύνολο των παρεχόμενων υπηρεσιών εμπλουτίζεται συνεχώς. Παρέχει και τα 3 μοντέλα ανάπτυξης (IaaS, PaaS, SaaS).

4.2.5 Google Cloud Platform

Σουίτα υπηρεσιών cloud computing που εκτελούνται στην υποδομή/υλικό που χρησιμοποιεί η ίδια η Google για τις υπηρεσίες της. Περιλαμβάνει συστατικά για υπηρεσίες υπολογιστικής ισχύος, αποθηκευτικό χώρο, data analytics καθώς και machine learning. Προσφέρονται και τα 3 μοντέλα ανάπτυξης (IaaS, PaaS, SaaS). Η διαχείριση γίνεται μέσω ειδικών συστατικών/αρθρωμάτων.

4.2.6 OpenNebula

Απλή αλλά ισχυρή λύση για τη δημιουργία και διαχείριση clouds και data centers. Χρησιμοποιείται για επιστημονικούς υπολογισμούς και διαχείριση big data. Παρέχει ενιαίο περιβάλλον διαχείρισης για private και public πόρους ακολουθώντας το μοντέλο υλοποίησης IaaS.

4.3 Cloud Management Platforms

Το λογισμικό το οποίο παρέχει στους πελάτες των cloud υπηρεσιών δυνατότητες για τη δημιουργία, τη διαχείριση και την παρακολούθηση των πόρων του cloud ονομάζεται Λογισμικό Διαχείρισης Cloud (CMP – Cloud Management Platform) [27]. Μάλιστα σε κάποια γλωσσάρια – όπως αυτό της εταιρείας Gartner (https://www.gartner.com/it-glossary/cloud-management-platforms) – καθορίζονται ελάχιστες απαιτήσεις που πρέπει να πληρούνται από τα προϊόντα ώστε να εμπίπτουν σε αυτή την κατηγορία. Αυτές είναι η παροχή έτοιμων εικόνων συστημάτων (system images), οι δυνατότητα μέτρησης και χρέωσης συγκεκριμένων υπηρεσιών με συγκεκριμένες μονάδες (αποθηκευτικός χώρος, επεξεργαστική ισχύς, εύρος ζώνης) και η παραμετροποίηση της διαχείρισης του φόρτου εργασίας. Στις μεγαλύτερες Cloud Platforms, το CMP αποτελεί ξεχωριστό συστατικό – συνήθως με τη μορφή αρθρώματος - το οποίο έχει σκοπό την παρακολούθηση και διαχείριση των υπολοίπων (συστατικών).

4.4 Virtualization

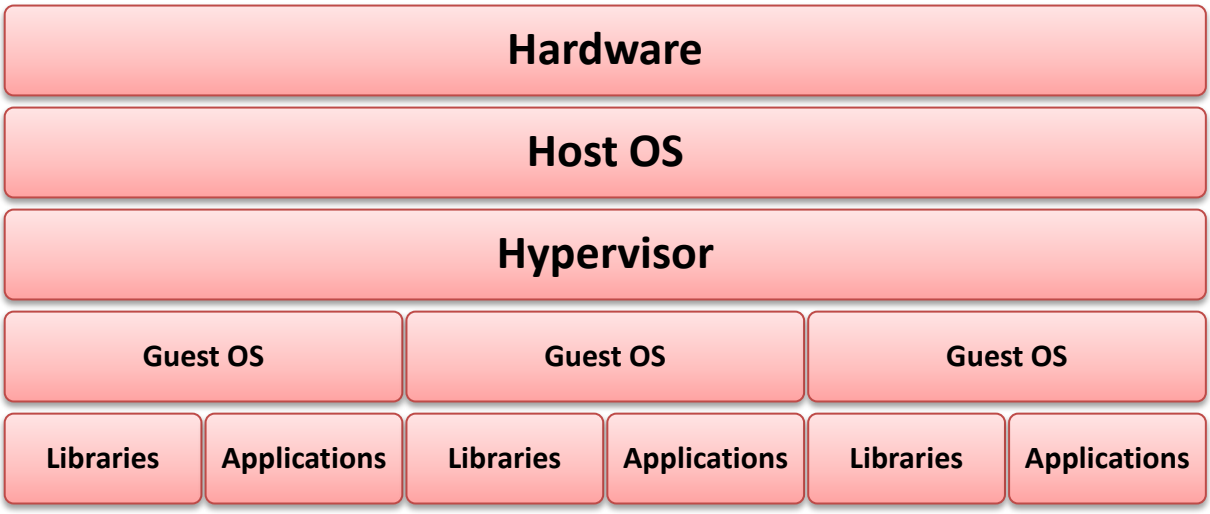
Ο όρος Virtualization αναφέρεται στη δημιουργία και διαχείριση εικονικών πόρων - μη πραγματικών. Οι πιο συνηθισμένοι εικονικοί πόροι είναι συγκεκριμένες πλατφόρμες και αρχιτεκτονικές επεξεργαστών, αποθηκευτικά μέσα και δικτυακές συσκευές. Παρόλο που η τεχνολογία virtualization είναι αρκετά παλιά, η χρήση της στην τεχνολογία cloud είναι νέα. Η χρήση του virtualization σε συνδυασμό με το cloud δίνει τη δυνατότητα στους παρόχους των υπηρεσιών cloud να δημιουργήσουν μόνοι τους, ακριβώς στα μέτρα τους την επιθυμητή υποδομή, συνδυάζοντας εικονικούς πόρους. Παρόλο που πολλές φορές τα όρια δεν είναι απολύτως ξεκάθαρα, οι βασικές κατηγορίες του Virtualization σύμφωνα με τη Wikipedia [28] είναι οι ακόλουθες :

- Hardware/Platform Virtualization
    - Full Virtualization
    - Paravirtualization
- Desktop Virtualization
- Software Virtualization
- Memory Virtualization
- Storage Virtualization
- Data Virtualization
- Network Virtualization

Μία διαφορετική προσέγγιση του Hardware virtualization αποτελεί το Hardware-assisted virtualization. Πρόκειται για μία μορφή full virtualization χρησιμοποιώντας τις δυνατότητες του υλικού του host computer και κυρίως των CPUs του. Ουσιαστικά επιτυγχάνεται virtualization ενός πλήρους συστήματος από πλευράς υλικού, χρησιμοποιώντας την ίδια αρχιτεκτονική και το ίδιο σετ εντολών του επεξεργαστή με αποτέλεσμα την εκτέλεση του guest λειτουργικού συστήματος σε περιβάλλον απομόνωσης από το host λειτουργικό σύστημα. Ακολουθούν γνωστές υλοποιήσεις/πλατφόρμες του hardware-assisted virtualization [29] [30]:

- Kernel-based Virtual Machine (KVM) – Ο πυρήνας του Linux με την προσθήκη των κατάλληλων modules μετατρέπεται σε λογισμικό hypervisor (με προϋπόθεση την υποστήριξη του virtualization από τη CPU). Το KVM αποτελεί ιδανική επιλογή για συστήματα υψηλών επιδόσεων
- Vmware Workstation – Hosted hypervisor με δυνατότητα χρήσης Windows και Linux συστημάτων ως host.
- Vmware Fusion – Σχεδιασμένο αποκλειστικά για Mac, δίνοντας τη δυνατότητα σε αυτά που διαθέτουν επεξεργαστή Intel να τρέξουν εικονικές μηχανές Microsoft Windows, Linux, Solaris κλπ.

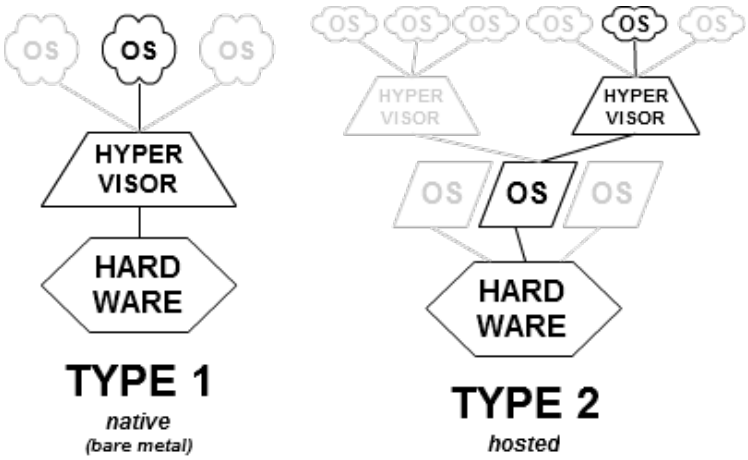
- Hyper-V – Προϊόν της Microsoft, τύπου native, χρησιμοποιείται για τη διαχείριση εικονικών μηχανών σε συστήματα Windows.
- Xen – Λογισμικό που επιτρέπει σε εικονικές μηχανές διαφορετικών συστημάτων (x86, x86-64, ARM) να εκτελούνται ταυτόχρονα και να διαμοιράζονται το υλικό.
- Parallels Desktop for Mac – Virtualization για διαχείριση εικονικών μηχανών windows σε Mac με Intel επεξεργαστή.
- Oracle VM Server for SPARC – Εξειδικευμένο λογισμικό virtualization για επεξεργαστές SPARC V9 με σύστημα host το Solaris.
- VirtualBox – Λογισμικό virtualization από την Oracle, ΕΛΛ/ΛΑΚ, με μεγάλη υποστήριξη σε Host και Guest συστήματα όπως Linux, Windows, MacOS, FreeBSD, SOLARIS κλπ.
- PowerVM – Virtualization Software από την IBM, αναπτύχθηκε για τους επεξεργαστές της οικογένειας POWER με Host OS το PowerVM και Guest OS το Linux PowerPC .



Σχήμα 5 - Virtualization

4.4.1 Hypervisor / Virtual Machine Monitor

Το λογισμικό Hypervisor ή Virtual Machine Monitor είναι υπεύθυνο για τη δημιουργία, την εκτέλεση και τη διαχείριση των εικονικών μηχανών. Ένα σύστημα Hypervisor μπορεί να διατηρεί πολλά και διαφορετικά στιγμιότυπα εικονικών συστημάτων. Συχνά αναφέρεται και ως *host machine* εννοώντας το υλικό, το λειτουργικό σύστημα και το λογισμικό hypervisor. Το σύστημα hypervisor χωρίζεται σε 2 κατηγορίες



Σχήμα 6 – Κατηγορίες Hypervisor Software [31]

- Native/bare metal – Το λογισμικό hypervisor "τρέχει" απευθείας πάνω στο υλικό εκτελώντας χρέη λειτουργικού συστήματος και διαχείρισης των εικονικών μηχανών. Τέτοιο λογισμικό είναι το Oracle VM Server for SPARC και το Vmware ESX.
- Hosted – Υπάρχει λειτουργικό σύστημα για την διαχείριση του υλικού. Το λογισμικό hypervisor τρέχει ως ξεχωριστή εφαρμογή του λειτουργικού συστήματος. Γνωστά hosted λογισμικά είναι το VMware Workstation, Vmware Player, VirtualBox κλπ.

Κεφάλαιο 5
Cloud Forensics

Η βασική αρχιτεκτονική του cloud δεν είναι ιδιαίτερα φιλική προς τα forensics. Ο εντοπισμός και η ανάλυση οποιουδήποτε συμβάντος απαιτεί τη χρήση γρήγορων εργαλείων αλλά πρωτίστως εργαλείων με δυνατότητες περιορισμού του εύρους των ερευνών.

Τα εργαλεία cloud forensics έχουν να αντιμετωπίσουν πολλαπλές προκλήσεις [2]. Ειδικότερα ως προς τα αρχεία καταγραφής υπάρχει δυσκολία στα κριτήρια με τα οποία θα καθοριστεί ποια logs είναι χρήσιμα και ποια μπορούν να χρησιμοποιηθούν. Επιπρόσθετα, δεν υπάρχει κάποιο συγκεκριμένο στάνταρντ για το format της καταγραφής από τον CSP.

Ανεξάρτητα από το μοντέλο παροχής των υπηρεσιών (IaaS, PaaS, SaaS), τα δεδομένα που έχουν αξία για τον ερευνητή ενός συμβάντος στο cloud, δε βρίσκονται στην πλευρά του πελάτη. Σε αυτή βρίσκονται μόνο κάποια προσωρινά αρχεία, cookies και session information files. Αυτά δεν παρέχουν αρκετές πληροφορίες για το συσχετισμό μίας δραστηριότητας με περισσότερα από 3 από τα 7 Ws.

Γίνεται αντιληπτό πως η ανάλυση στην πλευρά του παρόχου είναι μονόδρομος. Θεωρώντας ως δεδομένη την εμπιστοσύνη προς την οντότητα του παρόχου, ένα εργαλείο και κατ' επέκταση ο ερευνητής πρέπει να μπορεί να ζητά από τον πάροχο συγκεκριμένα δεδομένα βάσει κριτηρίων. Η λήψη τους προϋποθέτει την ύπαρξη ασφαλούς διαύλου επικοινωνίας και η επεξεργασία τους σεβασμό στην ιδιωτικότητα των υπολοίπων χρηστών. Η απάντηση στα ανωτέρω ζητήματα ακούει στο όνομα Web API και API Endpoints. Η παροχή αυτών των διεπαφών είναι πλέον διαθέσιμη εξ' ορισμού σε κάθε cloud platform.

Σε μία δικανική έρευνα είναι απαραίτητη η ελαχιστοποίηση της συμμετοχής του CSP. Στην πράξη βέβαια είναι αδύνατο, ωστόσο η χρήση ανοιχτών προτύπων – όπως το CADF – μπορεί να περιορίσει τις δυνατότητες παρέμβασης στα αρχεία καταγραφής από τον CSP.

Το 2016, οι Naaz και Ahmad, μελέτησαν τις δυνατότητες και τους περιορισμούς των εργαλείων FROST και UFED Cloud Analyzer [32], ενώ την ίδια χρονιά δημοσιεύτηκε μελέτη ερευνητών που περιέγραφε τις εμπειρίες και τις δυσκολίες που αντιμετώπισαν κατά την ανάπτυξη 3 εργαλείων cloud forensics για το μοντέλο SaaS [33]. Τα εργαλεία αυτά είναι τα kumodd, kumodocs και kumofs.

Όπως διαπιστώνεται από ομάδα ερευνητών, το 2016, λόγω της απουσίας cloud forensics tools, οι ερευνητές χρησιμοποιούν τις υπάρχουσες λύσεις λογισμικού. Είτε εξετάζουν με αυτές τα αποκτηθέντα αρχεία – θεωρώντας τα εξ' ορισμού αξιόπιστα – είτε προσπαθούν να πραγματοποιήσουν εξ' αποστάσεων ανάλυση. Τα εργαλεία που παρέχουν τη δυνατότητα απομακρυσμένης ανάλυσης δεν έχουν όμως ελεγχθεί και πιστοποιηθεί σχετικά με την ορθότητα και τη διατήρηση της ακεραιότητας των αρχείων και δεν παρέχουν έτσι καμία διαβεβαίωση αποδοχής από τις δικαστικές αρχές [23].

Τέλος, υπάρχουν και custom λύσεις με κατεύθυνση το cloud που υλοποιούνται από οργανισμούς για χρήση στο εσωτερικό τους – μία τέτοια περίπτωση είναι το Netflix (https://www.netflix.com/). Αυτές οι λύσεις είτε είναι ιδιοταγείς και δεν διατίθενται στο ευρύ κοινό, είτε γίνονται εμπορικά προϊόντα είτε στην καλύτερη περίπτωση ανήκουν στην κατηγορία ΕΛΛ/ΛΑΚ.

5.1 Cloud Forensics Tools

Το λογισμικό που χρησιμοποιείται κατά τη διαδικασία των forensics παράγει έξοδο και οδηγεί σε συμπεράσματα με σκοπό να γίνουν αποδεκτά από τις δικαστικές αρχές. Απαραίτητη προϋπόθεση είναι η διαβεβαίωση πως το λογισμικό δεν παραποιεί τα εξεταζόμενα δεδομένα. Επίσης είναι αναγκαίο να λειτουργεί με τέτοιο τρόπο ώστε να έχει τη δυνατότητα οποιοσδήποτε ερευνητής ακολουθώντας συγκεκριμένα βήματα να παράγει την ίδια έξοδο.

Το ινστιτούτο NIST εφαρμόζει μία διαδικασία αξιολόγησης των εργαλείων forensics ως προς τα ανωτέρω ζητήματα. Η αξιολόγηση περιλαμβάνει μία σειρά από ελέγχους και δοκιμές που ονομάζεται Computer Forensic Tool Testing (CFTT) και παράγει αποτελέσματα με τη μορφή αναφορών (CFTT Reports). Με αυτόν τον τρόπο δημιουργείται ένας κατάλογος εργαλείων, χωρισμένα σε κατηγορίες (disk imaging, file carving, email parsing), εγκεκριμένων από το NIST (https://toolcatalog.nist.gov/) . Αυτή η έγκριση παίζει καταλυτικό ρόλο ως προς την αποδοχή των αποτελεσμάτων του εργαλείου από της δικαστικές αρχές. Οι ερευνητές σαφώς προτιμούν τη χρήση των εργαλείων που εγκρίνονται από το NIST.

Ωστόσο, η αναζήτηση στον κατάλογο για εργαλεία cloud forensics φέρνει στην επιφάνεια την έλλειψη σε αυτόν τον τομέα. Μόλις 6 εργαλεία σχετίζονται με τα cloud forensics (https://toolcatalog.nist.gov/populated_taxonomy/index.php?all_tools=all&ff_id=20&1%5B%5D=any&2%5B%5D=any).

- Belkasoft Evidence Center (https://belkasoft.com/ec)
- Elcomsoft Cloud Explorer (https://www.elcomsoft.com/ecx.html)
- Elcomsoft Phone Breaker (https://www.elcomsoft.com/eppb.html)
- Internet Evidence Finder (IEF) (https://www.magnetforensics.com/magnet-ief/)
- Magnet AXIOM (https://www.magnetforensics.com/magnet-axiom/)
- UFED Cloud Analyzer (https://www.cellebrite.com/en/products/ufed-cloud-analyzer/)

Η σειρά είναι αλφαβητική και δεν αποτελεί ταξινόμηση. Σημαντική παρατήρηση είναι πως όλα αφορούν υπηρεσίες SaaS.

Ακολουθεί σύντομη επισκόπηση 2 από τα 6 παραπάνω εμπορικά εργαλεία και 3 εργαλείων που δημιουργήθηκαν στα πλαίσια ακαδημαϊκής έρευνας.

5.1.1 MAGNET AXIOM Cloud & Oxygen Forensic Suite

Το εργαλείο Magnet Axiom Cloud χρησιμοποιείται κυρίως σε υποδομές SaaS. Δημιουργήθηκε από την εταιρεία Magnet Forensics (https://www.magnetforensics.com/) και είναι εμπορικό λογισμικό. Είναι εγκεκριμένο από τον οργανισμό NIST. Έχει επιλογή λήψης των αρχείων ενός χρήστη είτε από δημόσιο cloud είτε από ιδιωτικό – εισάγοντας τα διαπιστευτήρια του χρήστη. Δεδομένα μπορεί να αντλήσει (στην τρέχουσα έκδοση 1.2) από εφαρμογές των Google, Apple, Facebook, Microsoft, Twitter. Τα ληφθέντα αρχεία μπορεί να προέλθουν και από πηγές όπως κινητά τηλέφωνα ή υπολογιστές.

Η σουίτα Oxygen Forensic αποτελεί προϊόν της εταιρείας Oxygen Forensics Inc. (https://www.oxygen-forensic.com/). Είναι προσανατολισμένο κυρίως στις φορητές συσκευές όπως τα κινητά τηλέφωνα και συλλέγει δεδομένα από αυτές και από εφαρμογές cloud που εκτελούν. Υποστηρίζει συστήματα Android, iPhone και BlackBerry.

Περιορισμοί και αδυναμίες

Και τα 2 εργαλεία περιορίζουν τη λειτουργικότητά τους στην αρχιτεκτονική SaaS και μάλιστα σε πολύ μικρό και συγκεκριμένο αριθμό εφαρμογών cloud. Επιπρόσθετα, έχουν κόστος απόκτησης και εκπαίδευσης για τον ερευνητή.

5.1.2 UFED Cloud Analyzer

Το εργαλείο UFED Cloud Analyzer είναι ιδιοταγές, με δυνατότητες εξόρυξης δεδομένων από κοινωνικά δίκτυα. Είναι εγκεκριμένο από το NIST. Αφορά κυρίως IoT συσκευές και βασίζει την αναζήτησή του σε πεδίο κλειδί – συνήθως κάποιο username. Προκειμένου να δημιουργήσει Timeline με τα δεδομένα που συλλέγει εφαρμόζει διαδικασία κανονικοποίησης μετατρέποντας τα σε ενιαία μορφή. Το εργαλείο κάνει χρήση APIs ώστε να λάβει στιγμιότυπα δεδομένων από το cloud.

Περιορισμοί και αδυναμίες

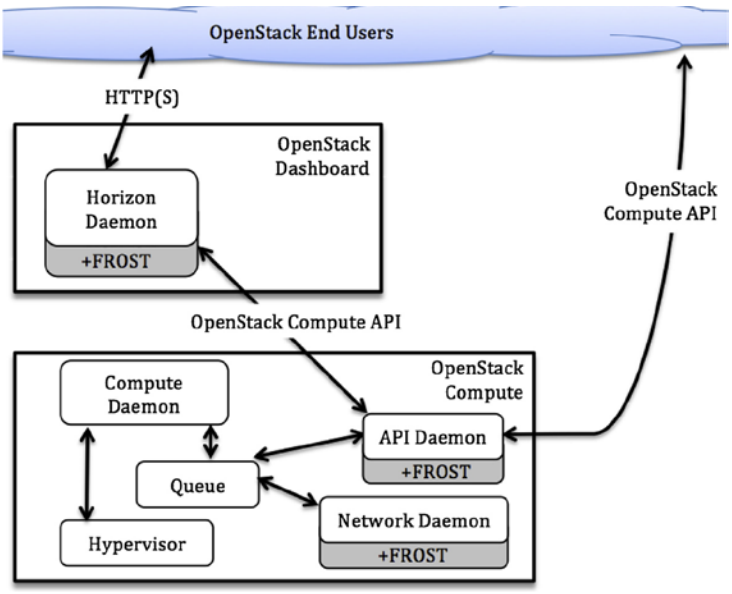
Στο μεγάλο όγκο δεδομένων του cloud και των κοινωνικών δικτύων ο εντοπισμός της χρήσιμης πληροφορίας είναι εργασία απαιτητική σε υπολογιστική ισχύ και χρονοβόρα. Η λύση του φιλτραρίσματος βάσει κριτηρίων εγκυμονεί σοβαρούς κινδύνους παραβίασης των εκάστοτε κανονισμών διατήρησης και διαχείρισης των προσωπικών δεδομένων.

5.1.3 Forensic OpenStack Tools (FROST)

Το εργαλείο FROST αφορά αποκλειστικά την πλατφόρμα OpenStack. Πρόκειται για λογισμικό λήψης και αποθήκευσης κυρίως αρχείων καταγραφής σε δενδροειδή μορφή μέσω API και υλοποιήθηκε σε 2 συστατικά του OpenStack. Στο Nova[6] project και στο Horizon [7]project. Η διαχείριση του FROST γίνεται μέσω του περιβάλλοντος διαχείρισης του ίδιου του cloud. Δεν έχει καμία αλληλεπίδραση με τις εικονικές μηχανές και το τι συμβαίνει στο εσωτερικό τους (λειτουργικό σύστημα, εφαρμογές και δεδομένα). Με αυτόν τον τρόπο εξασφαλίζεται πως ακόμα και ένα παραβιασμένο ή μολυσμένο από κακόβουλο λογισμικό σύστημα δεν επηρεάζει τα δεδομένα που συλλέγει το FROST.

[6] Συστατικό υπεύθυνο για τη δημιουργία εικονικών μηχανών

[7] Συστατικό παροχής πίνακα ελέγχου και διαχείρισης των υπόλοιπων συστατικών και την υποδομής



Σχήμα 7 – Τρόπος λειτουργίας FROST - αλληλεπίδραση μέσω API [34]

Συγκεκριμένα, οι Dykstra και Sherman, το 2013 κατά το σχεδιασμό και την υλοποίηση του FROST [34] έθεσαν κάποιες προϋποθέσεις που πρέπει να πληρούν τα εργαλεία cloud forensics:

- Συμβατότητα με υπάρχοντα forensics formats (DFXML[8])
- Εύκολη υλοποίηση και ενσωμάτωση στο cloud
- Ανοιχτότητα και επεκτασιμότητα
- Δυνατότητα κλιμάκωσης
- Συμμόρφωση με πρακτικές και απαιτήσεις τις αγοράς

Περιορισμοί και αδυναμίες

Το FROST απαιτεί την ύπαρξη εμπιστοσύνης στον CSP. Εκεί εκτελείται το host OS και δημιουργούνται και φυλάσσονται τα αρχεία καταγραφής. Η διατήρηση των δεδομένων, σε επίπεδο logs αλλά κυρίως σε επίπεδο virtual disks αποτελεί ζήτημα που πρέπει να λαμβάνει υπόψη το χρόνο ολοκλήρωσης της έρευνας.

Κατά το χρονικό διάστημα της παρουσίασης του FROST, το OpenStack βρισκόταν ακόμα σε πρώιμο στάδιο. Η ωρίμανση του λογισμικού αλλά κυρίως η συνεργασία του FROST με τα υπόλοιπα βασικά συστατικά του OpenStack – όπως KeyStone[9], Swift[10] κλπ – ήταν επιτακτική.

[8] XML σχήμα για παροχή διαλειτουργικότητας ανάμεσα σε διαφορετικά εργαλεία forensics
[9] Συστατικό υπεύθυνο για τη διαχείριση ταυτότητας στο OpenStack
[10] Συστατικό διαχείρισης cloud storage

Ωστόσο, η ιδέα του FROST αποδείχθηκε εξαιρετική στη σύλληψη της ώστε ενσωματώθηκε στο OpenStack με τη μορφή ενός ξεχωριστού project, του Ceilometer. Το Ceilometer παρέχει υπηρεσίες τηλεμετρίας με τη μορφή του προτύπου CADF μέσω API endpoints.

5.1.4 kumodd/kumodocs/kumofs

Τα 3 εργαλεία απαρτίζουν μία cloud forensics σουίτα για συγκεκριμένους providers. Κάθε ένα από αυτά τροφοδοτεί με δεδομένα το επόμενο

- kumodd – cloud drive acquisition tool βασισμένο στο API του μελετώμενου CSP
- kumodocs – εργαλείο ανάλυσης Google Docs
- kumofs – filesystem για cloud data

Τα δεδομένα λαμβάνονται χρησιμοποιώντας τα APIs 2 Web Applications – του Google Drive και του Microsoft Dropbox. Χαρακτηριστικό του βαθμού διαφοροποίησης από πάροχο σε πάροχο τόσο σε θέματα σχεδιασμού όσο και στα προσφερόμενα μέσω API δεδομένα είναι το εξής :

- Microsoft Dropbox – 18 metadata attributes
- Google Drive – 100 metadata attributes

Περιορισμοί και αδυναμίες

Το 2016, ερευνητές διαπίστωσαν την έλλειψη συνολικής λύσης και την ανυπαρξία ανοιχτής πλατφόρμας που θα επιτρέπει στην κοινότητα των χρηστών και ερευνητών το διαμοιρασμό μεθόδων και τεχνογνωσίας. Εξετάζοντας 2 μόνο από τα πολλά cloud based web application καταλήγουν δυστυχώς στο συμπέρασμα πως κάθε περίπτωση απαιτεί εξειδικευμένα για αυτήν εργαλεία [33].

5.1.5 Diffy

Το εργαλείο ανοιχτού κώδικα diffy (https://github.com/Netflix-Skunkworks/diffy) δημιουργήθηκε από την ομάδα Security Intelligence and Response Team του NetFlix. Αυτοπροσδιορίζεται ως DFIR (Digital Forensics and Incident Response) εργαλείο το οποίο μπορεί να εντοπίσει γρήγορα συμβάν ασφαλείας σε εικονικές μηχανές Linux οι οποίες φιλοξενούνται από το Amazon Web Services cloud. Στην πράξη συγκρίνει στιγμιότυπα των εξεταζόμενων εικονικών μηχανών με μία εικονική μηχανή σημείο αναφοράς προκειμένου να εντοπίσει ύποπτες αλλαγές. Η αρχιτεκτονική του είναι αρθρωτή. Σύμφωνα με το blog του NetFlix [35], υποστηρίζονται 2 τρόποι εντοπισμού των αλλαγών, η “functional baseline” και η “clustering” μέθοδος.

Περιορισμοί και αδυναμίες

Είναι εργαλείο με στόχο αποκλειστικά το use case του φορέα από τον οποίο αναπτύχθηκε. Δεν είναι πιστοποιημένο από τον οργανισμό NIST. Όπως αναφέρει η ομάδα ανάπτυξης,

πρόκειται για project τύπου “Skunkworks”, δηλαδή με συνεχή ανάπτυξη αλλά χωρίς δυνατότητα παροχής υποστήριξης.

Κεφάλαιο 6
Cloud Auditing

Το DMTF ξεκίνησε μία προσπάθεια με όνομα "*Cloud Management Initiative*" η οποία έχει σκοπό τη δημιουργία και το συνδυασμό προτύπων στο cloud. Ο στόχος είναι η ενοποίηση και η διαλειτουργικότητα της διαχείρισης της υποδομής cloud από τους παρόχους, τους πελάτες, τους προγραμματιστές και τις κοινότητες. Οι ομάδες εργασίες μέχρι και σήμερα είναι οι :

- Cloud Management Working Group (CMWG)
- Cloud Auditing Data Federation Working Group (CADF WG)
- Software Entitlement Working Group (SEWG)
- Open Virtualization Format Working Group (OVF)

Κάθε ομάδα εργασίας είναι επιφορτισμένη με τη μελέτη και την πρόταση συγκεκριμένων προτύπων. Μέλη του DMTF είναι μερικές από τις κορυφαίες εταιρείες στο χώρο της Πληροφορικής και των Τηλεπικοινωνιών καθώς και πολλά πανεπιστήμια και ερευνητικά ιδρύματα. Ενδεικτικά, μέλη είναι οι :

- Intel
- Lenovo
- Dell
- Huawei
- Oracle
- Cisco κλπ.

6.1 Cloud Auditing

Η διαδικασία του Cloud Auditing (κατά λέξη "Έλεγχος του Cloud" – η μετάφραση στα ελληνικά δεν αποδίδει το σωστό νόημα) είναι συνεχής και έχει σκοπό να μετρήσει και να διαθέσει στους παρόχους υπηρεσιών cloud στοιχεία σχετικά με την απόδοση και τη συμμόρφωση με τις απαιτήσεις ασφαλείας των υπηρεσιών τους. Πρόκειται για μετρικές και στατιστικά με βάσει τα οποία οι εταιρείες παροχής υπηρεσιών cloud έχουν τη δυνατότητα να παρακολουθούν αλλά και να βελτιώνουν συνεχώς το επίπεδο των υπηρεσιών τους και το επίπεδο ασφαλείας τους.

Το Cloud Auditing διενεργείται συνήθως από μία τρίτη οντότητα – αν δεν υπάρχει ειδική ομάδα ή τμήμα εντός του οργανισμού – η οποία αναλαμβάνει να ολοκληρώσει σε στάδια, ανάλογα με τη στρατηγική της οντότητας, τον έλεγχο του παρόχου ως προς την ασφάλεια των επικοινωνιών, τη διαχείριση και αναβάθμιση των συστημάτων, τη διαχείριση των δεδομένων, τη διαχείριση επικινδυνότητας και τη διαχείριση και αντιμετώπιση των συμβάντων.

Ωστόσο είναι σημαντικό να επισημανθεί το γεγονός πως πολλές φορές προς όφελος του Management των παρόχων, το auditing επικεντρώνεται στην απόδοση των υπηρεσιών των συστημάτων και όχι στην επακριβή αποτύπωση του τι ακριβώς συμβαίνει στα συστήματα σε τεχνικό επίπεδο. Το auditing πρέπει να περιλαμβάνει διοίκηση, τεχνικό προσωπικό και υποδομές [36]. Είναι ανάγκη να εκτείνεται σε όλα τα επίπεδα (από Hardware, Host OS, Virtualization Software κλπ) καθώς και σε όλα τα μοντέλα ανάπτυξης (private, community, public, hybrid)

Οι πελάτες των παρόχων υπηρεσιών cloud θεωρούν αυτονόητη την πρόσβαση σε auditing και monitoring δεδομένα. Στους παρόχους αυτούς οι πελάτες θα εμπιστευθούν τα εταιρικά και προσωπικά τους δεδομένα και για αυτό απαιτούν συγκεκριμένες εγγυήσεις.

Συνοψίζοντας, το Cloud Auditing αφορά σε διαδικασίες ελέγχου που μπορεί να εκτείνονται κατακόρυφα στο οργανόγραμμα ενός οργανισμού καθώς και σε ολόκληρη την υποδομή του. Ειδικότερα όμως για το κάθε εξεταζόμενο τμήμα είναι επιτακτική η ανάγκη χρήσης μίας κοινής γλώσσας και εργαλειοθήκης. Μάλιστα, όπως επισημαίνει και ο Konoor κάθε λειτουργία στην υποδομή πρέπει να γίνεται με τέτοιο τρόπο που να παρέχει σχετικές με αυτή πληροφορίες για μελλοντική ανάλυση και επισκόπηση [37].

6.1.1 ISACA

Το Cloud Auditing εντάσσεται στο γενικό πλαίσιο του Information Systems Auditing. Ο ανεξάρτητος και μη κερδοσκοπικός οργανισμός ISACA (Information Systems Audit and Control Association) (https://www.isaca.org/) προτείνει διαδικασίες και πρακτικές κοινά αποδεκτές και συνεχώς εξελισσόμενες για τον έλεγχο των πληροφοριακών συστημάτων. Ένα υποσύνολο αυτών των διαδικασιών έχει εφαρμογή στα συστήματα cloud. Δημοσιεύσεις, μελέτες, αναλύσεις και οδηγοί αποτελούν βασική βοήθεια καθώς συμβουλεύουν τους ελεγκτές συστημάτων για τις συνεχώς μεταβαλλόμενες απαιτήσεις στον τομέα της ασφάλειας πληροφοριών.

6.2 Cloud Auditing Data Federation (CADF)

Το CADF standard δεν αποτελεί έτοιμο εργαλείο, βιβλιοθήκη ή middleware που μπορεί να χρησιμοποιήσει ένας CSP. Πρόκειται για ένα ολοκληρωμένο μοντέλο περιγραφής συμβάντων που σχετίζονται με τον έλεγχο της ασφάλειας των εφαρμογών που εκτελούνται στο cloud.Σε μία υποδομή cloud οι πόροι ανταλλάσσουν συνεχώς πληροφορίες μεταξύ τους. Συχνά δε, υπάρχει η ανάγκη η ανταλλαγή να γίνεται κάτω από ένα καθεστώς κανόνων. Το μοντέλο ορίζει αυστηρά την έννοια του συμβάντος (event).

Τύπος συμβάντος	**Επεξήγηση**
Actual Event	Οτιδήποτε συμβαίνει σε ένα σύστημα
Event Record	Η δομημένη απεικόνιση των πληροφοριών που σχετίζονται με το Actual Event
CADF Event Record	Event Record δομημένο με βάση το καθορισμένο CADF Event μοντέλο

Πίνακας 1 – Τύποι συμβάντων

Το CADF δίνει τη δυνατότητα κατηγοριοποίησης των συμβάντων κατά τύπο, ανεξάρτητα από το συστατικό του cloud (cloud application) ή το μοντέλο ανάπτυξης (private, hybrid κλπ). Τα σχετικά με συμβάντα δεδομένα συλλέγονται από όλα τα στρώματα του cloud (Application, Platform και Infrastructure). Τα δεδομένα που απαιτεί το CADF για τη δημιουργία της εγγραφής (CADF Event Record) δεν αποκαλύπτουν την αρχιτεκτονική ή τις τεχνικές λεπτομέρειες που θα μπορούσε κάποιος κακόβουλος χρήστης να εκμεταλλευθεί σχετικά με την υποδομή του CSP [38].

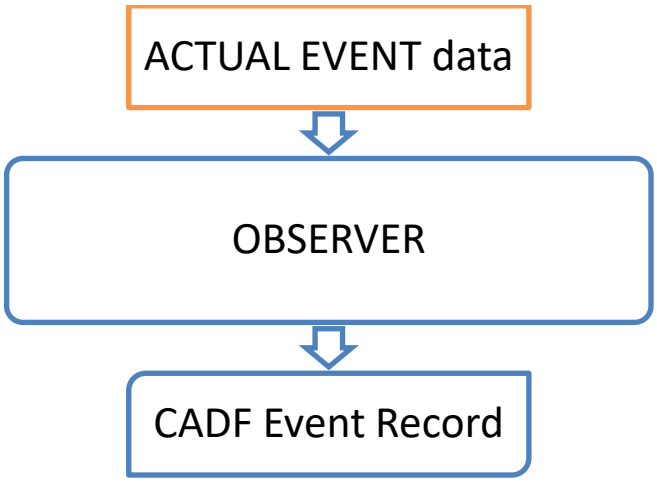
6.2.1 Επισκόπηση του CADF

Συστατικό	**Επεξήγηση**
OBSERVER	Ο πόρος που δημιουργεί το CADF Event Record και παρακολουθεί τους υπόλοιπους πόρους
INITIATOR	Ο πόρος που δημιούργησε την ενέργεια (ACTION) σύμφωνα με τον OBSERVER
ACTION	Η ενέργεια που εκτελέστηκε ή προσπάθησε να εκτελεστεί προς το στόχο (TARGET) σύμφωνα με τον OBSERVER
TARGET	Ο πόρος τον οποίο αφορούσε η ενέργεια (ACTION) από τον INITIATOR σύμφωνα με τον OBSERVER
OUTCOME	Το αποτέλεσμα της ενέργειας στο στόχο (TARGET)

Πίνακας 2 – Βασικά συστατικά του CADF Event Record

Ο OBSERVER παρακολουθεί έναν πόρο. Αν προκύψει κάποιο Event τότε δημιουργεί το CADF Event Record. Το CADF Record περιγράφει ποιος (INITIATOR) προσπάθησε να κάνει τι (ACTION) και σε ποιον (TARGET) και με ποιο αποτέλεσμα (OUTCOME).

Σχήμα 8 – Actual Event



Σχήμα 9 – Δημιουργία CADF Event Record

Ο OBSERVER είναι αναλαμβάνει το έργο της συνεχούς παρακολούθησης όλων των πόρων συμπεριλαμβανομένου και του εαυτού του (αφού και ο OBSERVER είναι και αυτός πόρος). Υπό συνθήκες απαιτούνται 2 επιπλέον πεδία για την απεικόνιση του CADF Event Record.

Συστατικό	**Επεξήγηση**
MEASUREMENT	Στατιστικά και μετρικές για τον TARGET
REASON	Πληροφορίες για την κατάταξη του OUTCOME

Πίνακας 3 – Υπό συνθήκη συστατικά του CADF Event Record

Εκτός από τα υποχρεωτικά συστατικά/πεδία του μοντέλου (Πίνακας 2) υπάρχουν και βοηθητικά πεδία (REPORTER, REPORTERCHAIN). Ειδικό τύπο αποτελεί το πεδίο *EventType*.

Η τιμή του EventType κατατάσσει το CADF Event Record σε συγκεκριμένη κατηγορία συμβάντων επηρεάζοντας παράλληλα τα υποχρεωτικά πεδία της εγγραφής.

Τιμή EventType	**Επεξήγηση**
Monitor	Παροχή πληροφοριών σχετικά με την κατάσταση και τις ιδιότητες πόρου
Activity	Παροχή πληροφοριών σχετικά με ενέργειες που σχετίζονται με τον πόρο
Control	Παροχή πληροφοριών σχετικά με τις επιπτώσεις κάποιας διαδικασίας ελέγχου σε έναν πόρο

Πίνακας 4 - Τύποι συμβάντων κατά το CADF

Ανεξαρτήτως όμως της τιμής του EventType, τα πιο σημαντικά πεδία του μοντέλου αποτυπώνονται στον ακόλουθο πίνακα.

Πεδίο	**Επεξήγηση**
typeURI	Δήλωση της έκδοσης του προτύπου με το οποίο περιγράφεται το Event
id	Αναγνωριστικό του Event
eventType	Ο τύπος του Event
eventTime	Η χρονοσήμανση που κατά τον OBSERVER ξεκίνησε ή συνέβη το Event
action	Η ενέργεια προς τον TARGET
outcome	Το αποτέλεσμα του ACTION
initiator	Ο πόρος που εκτέλεσε την ενέργεια
initiatorid	Αναγνωριστικό του INITIATOR
Target	Ο πόρος τον οποίο αφορούσε η ενέργεια
Targetid	Αναγνωριστικό του TARGET
Observer	Ο πόρος που παρακολουθεί τους υπόλοιπους και καταγράφει το συμβάν
Observerid	Αναγνωριστικό του OBSERVER
Measurement	Μετρήσεις και στατιστικά σχετικά με το Event
Reason	Περισσότερες πληροφορίες σχετικά με το OUTCOME
Name	Περιγραφικό όνομα για το Event
Severity	Σοβαρότητα του Event – έχει νόημα μόνο για τον OBSERVER
Duration	Διάρκεια του ACTION του Event
Tags	Ετικέτες για την περεταίρω κατάταξη του Event
Attachments	Περισσότερες πληροφορίες σχετικά με το Event
Reporterchain	Πληροφορίες σχετικά με την ακολουθία των χειρισμών από τους REPORTERs

Πίνακας 5 - Τα πεδία του CADF Event Record

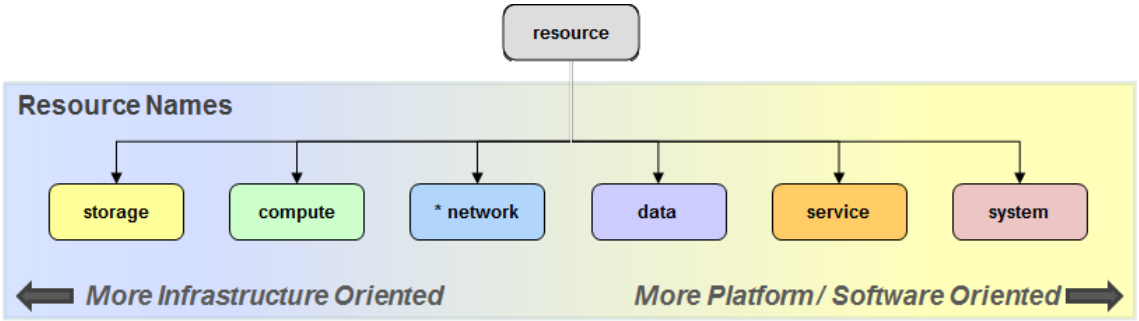
Πεδίο	Περιγραφή
Id	Αναγνωριστικό του πόρου
typeURI	Η κατηγορία/κατάταξη του πόρου
Name	Το όνομα του πόρου
Domain	Ο τομέας στον οποίο βρίσκεται ο πόρος
Credential	Διαπιστευτήρια σχετικά με την ταυτότητα του πόρου
Addresses	Διεύθυνση και URL του πόρου
Host	Πληροφορίες σχετικά με το σύστημα που υλοποιεί τον πόρο
Geolocation[11]	Γεωγραφική τοποθεσία του πόρου
Geolocationid	Αναγνωριστικό γεωγραφικής τοποθεσίας
Attachments	Πρόσθετες πληροφορίες σχετικά με τον πόρο και το περιεχόμενό του

Πίνακας 6 - Πεδία της οντότητας "Πόρος"

6.2.2 Ταξινομίες CADF

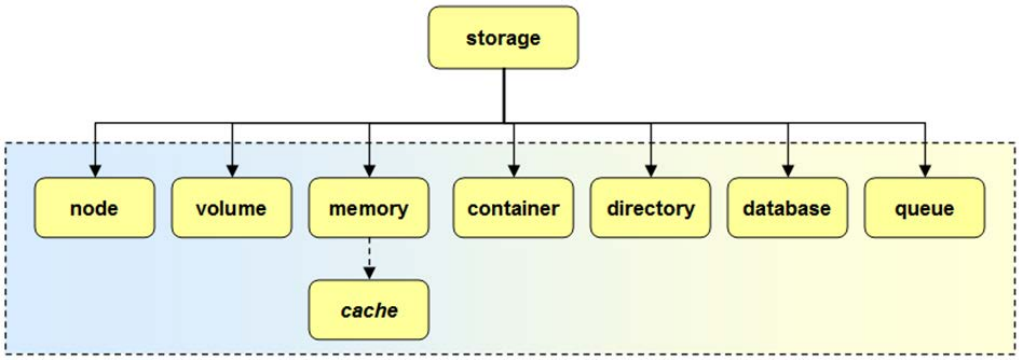
Η χρήση ενός πεδίου σε κάποιες περιπτώσεις δεν αρκεί για να αποτυπώσει την απαιτούμενη πληροφορία. Για αυτό το λόγο χρησιμοποιείται η κατηγοριοποίηση των τιμών σε ταξινομίες. Η οντότητα του πόρου (Resource) χρησιμοποιεί τις παρακάτω ταξινομίες:

- Storage
- Compute
- Network
- Data
- Service
- System
- Unknown

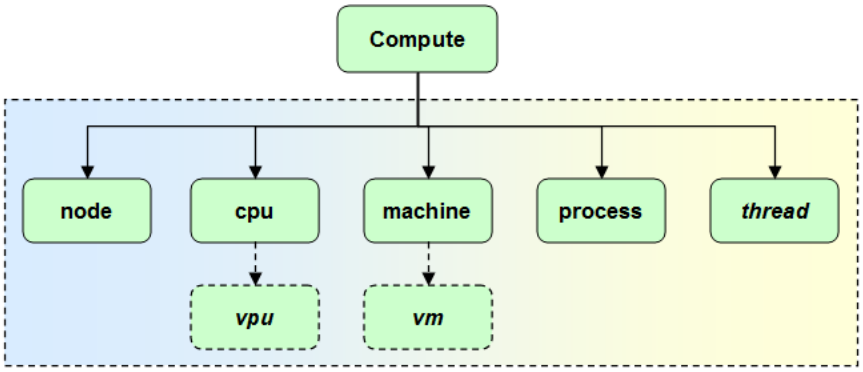


Σχήμα 10 – Ταξινομίες των πόρων [3]

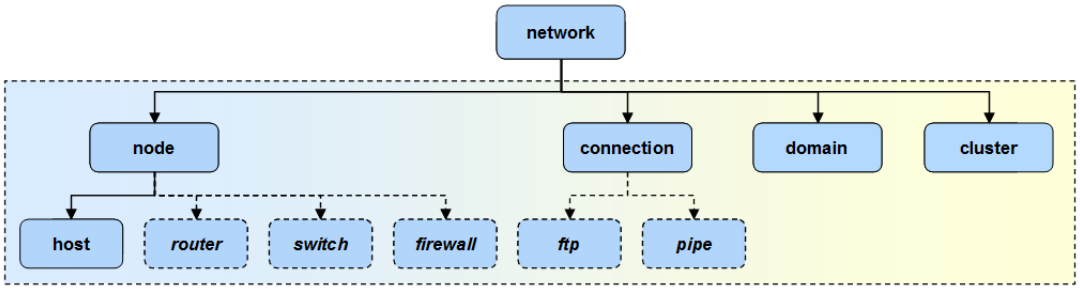
[11] Κατά το πρότυπο ISO-6709-2008 και με βάση τις οδηγίες του ICANN Final Implementation Plan for IDN cc TLD Fast Track Process



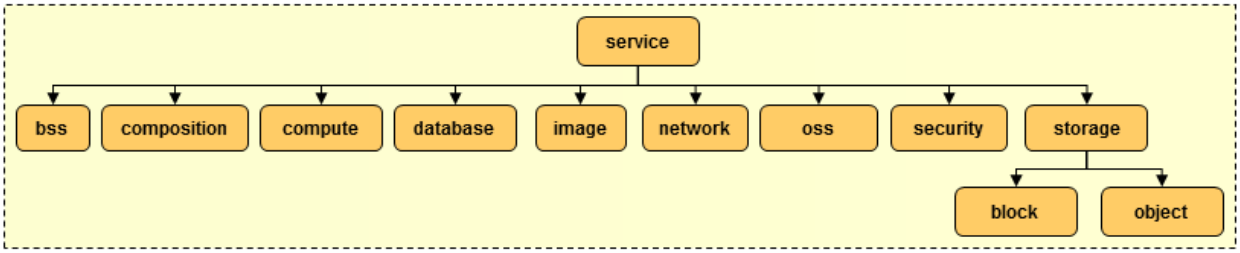
Σχήμα 11 – Ταξινομίες του πόρου τύπου “storage” [3]



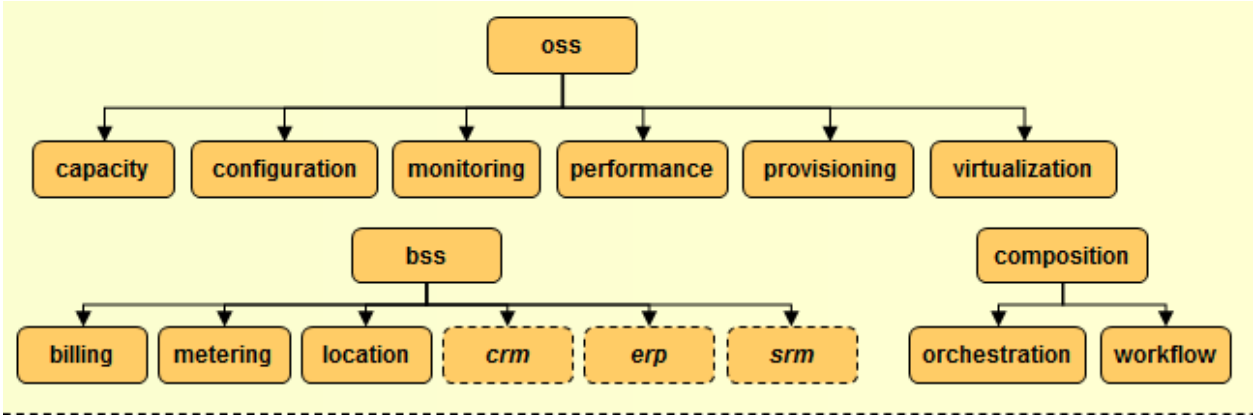
Σχήμα 12 – Ταξινομίες του πόρου τύπου “compute” [3]



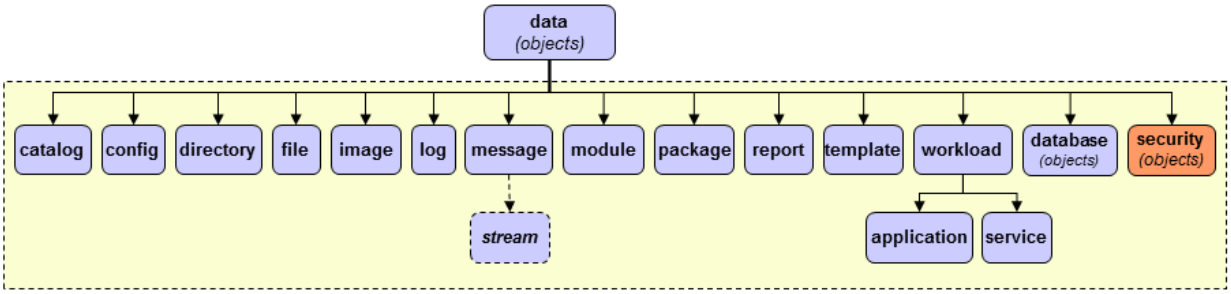
Σχήμα 13 - Ταξινομίες του πόρου τύπου “network” [3]



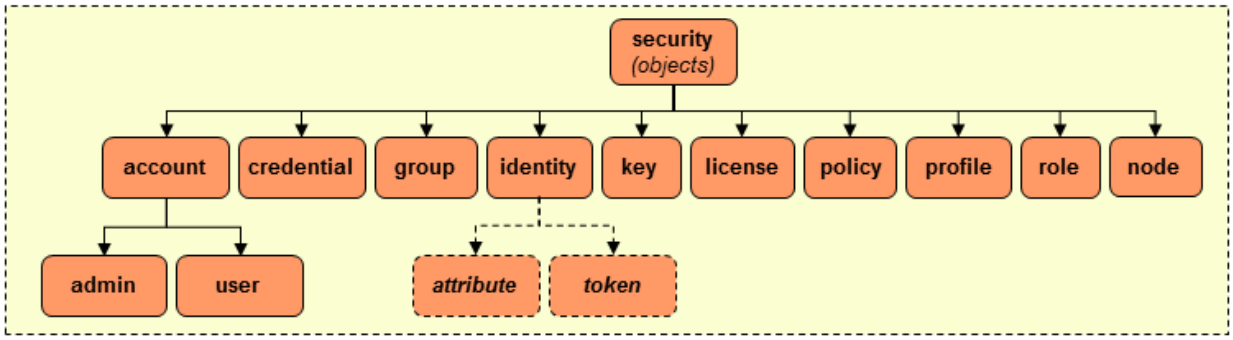
Σχήμα 14 - Ταξινομίες του πόρου τύπου “service” [3]



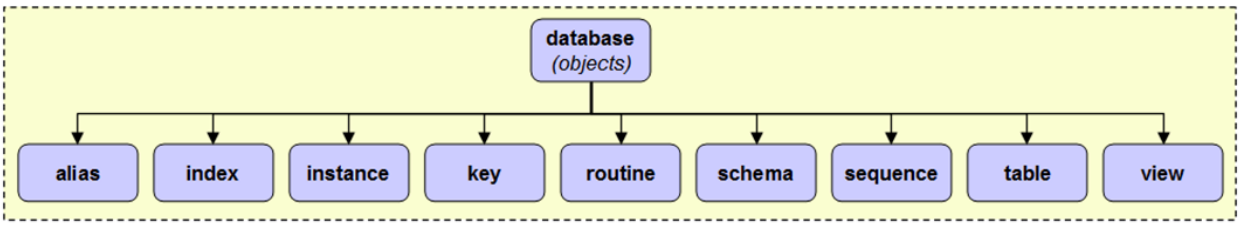
Σχήμα 15 - Ταξινομίες πόρων τύπου "service/oss","service/bss" ,"service/composition"[3]



Σχήμα 16 - Ταξινομίες του πόρου τύπου "data" [3]

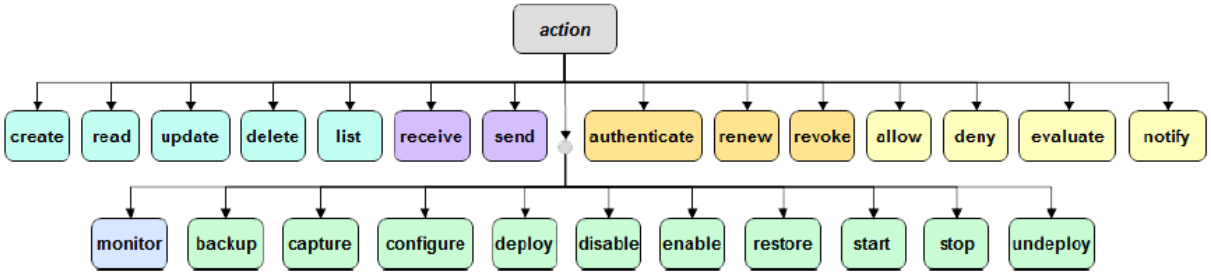


Σχήμα 17 - Ταξινομίες του πόρου τύπου "data/security" [3]

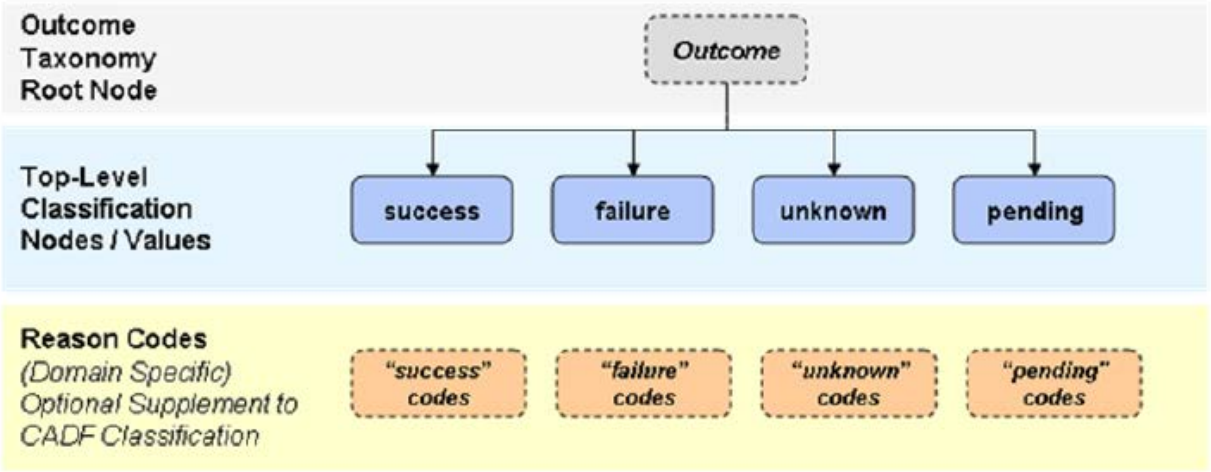


Σχήμα 18 - Ταξινομίες του πόρου τύπου "database" [3]

Αντίστοιχες ταξινομίες ακολουθούν οι οντότητες της ενέργειας "ACTION" και του αποτελέσματος "OUTCOME".



Σχήμα 19 - Ταξινομίες της οντότητας "ACTION" [3]



Σχήμα 20 - Ταξινομίες της οντότητας "OUTCOME" [3]

6.2.3 Σκοπός του CADF

Ο σκοπός του μοντέλου CADF είναι η παροχή πληροφοριών σχετικά με τη χρήση των πόρων στα περιβάλλοντα cloud κυρίως προς ερευνητές και ελεγκτές συστημάτων προκειμένου να βοηθηθούν στον εντοπισμό συγκεκριμένων δραστηριοτήτων. Δικαιολογημένα χαρακτηρίστηκε ως "CSI for clouds" [38]. Συγκεκριμένα, το μοντέλο CADF είναι σχεδιασμένο έτσι ώστε να δίνει απάντηση στα 7 W's of audit [39] [40].

W's	Περιγραφή	Σχετιζόμενα συστατικά
What	Τι δραστηριότητα παρουσιάστηκε και τι αποτέλεσμα είχε	EventType, ACTION, OUTCOME
When	Πότε εκτελέστηκε η δραστηριότητα	REPORTER
Who	Ποιος εκτέλεσε τη δραστηριότητα	INITIATOR
FromWhere	Από πού ξεκίνησε η δραστηριότητα	INITIATOR

OnWhat	Ποιος ήταν ο στόχος της δραστηριότητας	TARGET
Where	Πού καταγράφηκε η δραστηριότητα	OBSERVER
ToWhere	Πού βρίσκεται ο στόχος της δραστηριότητας	TARGET

Πίνακας 7 - 7 W's

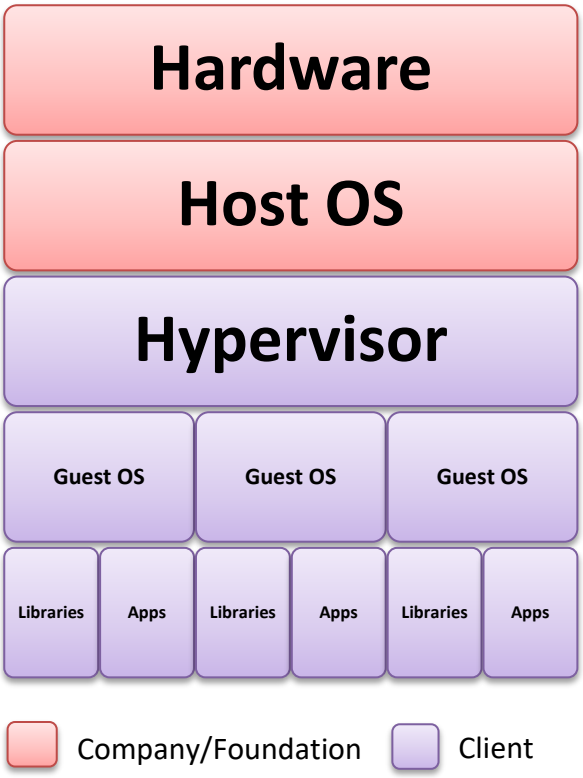
Κεφάλαιο 7
IaaS Landscape

Στην προσπάθεια για την αποτύπωση της τρέχουσας κατάστασης παγκοσμίως σχετικά με τις επιδόσεις, τις τάσεις αλλά και τις προτιμήσεις των επιχειρήσεων σε σχέση με τους CSPs γίνεται αισθητός ο διαχωρισμός ανάμεσα στις λύσεις που παρέχουν οι CSPs. Συγκεκριμένα ανάμεσα στις *ανοιχτού κώδικα* υλοποιήσεις και το ιδιόκτητο λογισμικό.

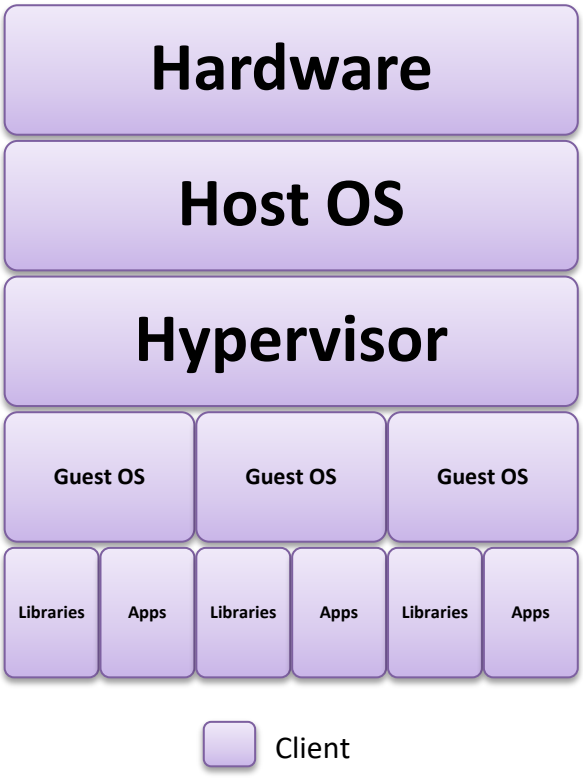
Οι ανοιχτού κώδικα λύσεις διατηρούν τα οφέλη της ανοιχτότητας και της παραμετροποίησης, ωστόσο υποχρεώνουν τον πελάτη – στις περισσότερες περιπτώσεις - να χρησιμοποιήσει το δικό του hardware και host os. Κάποιες λύσεις, υποστηριζόμενες ωστόσο από ιδρύματα ή κοινότητες χρηστών δίνουν στους πελάτες τους την επιλογή να φιλοξενήσουν το λογισμικό σε δικό τους υλικό ή σε κάποια συνεργαζόμενη εταιρεία.

Αντίθετα, οι ιδιοταγείς λύσεις απαλλάσσουν τον εταιρικό πελάτη από την ευθύνη απόκτησης του υλικού, την εγκατάσταση και διαχείριση του host os και όλα όσα αυτά συνεπάγονται (πχ αντίγραφα ασφαλείας των εικονικών μηχανών). Βέβαια, σε αυτές τις περιπτώσεις το μεγαλύτερο ίσως πρόβλημα είναι η άγνωστη μορφή προτύπων, αρχιτεκτονικών και ρυθμίσεων που χρησιμοποιούνται από τους δημιουργούς του cloud software. Ειδικότερα όμως στον τομέα του auditing και logging, θέματα που πραγματεύεται η παρούσα μεταπτυχιακή διατριβή, η τεκμηρίωση των πεδίων και των διαδικασιών είναι παραπάνω από επαρκής και για τις λύσεις ανοιχτού κώδικα και για τις ιδιοταγείς. Αυτό συμβαίνει διότι πλέον ο κάθε υποψήφιος πελάτης μίας πλατφόρμας cloud θέτει ως προϋπόθεση τη δυνατότητα απεριόριστης πρόσβασης σε auditing και logging δεδομένα. Μάλιστα είναι σύνηθες τα δεδομένα αυτά να απαιτεί να τα έχει στη διάθεσή του σε πραγματικό χρόνο προκειμένου να τα χρησιμοποιήσει ως είσοδο σε λογισμικό SIEM[12].

[12] Λογισμικό ανάλυσης σε πραγματικό χρόνο συμβάντων σχετικά με την ασφάλεια από υλικό, λειτουργικό σύστημα και εφαρμογές



Σχήμα 21 –Υλικό και host os υλοποιούνται σε εγκαταστάσεις εκτός ευθύνης πελάτη



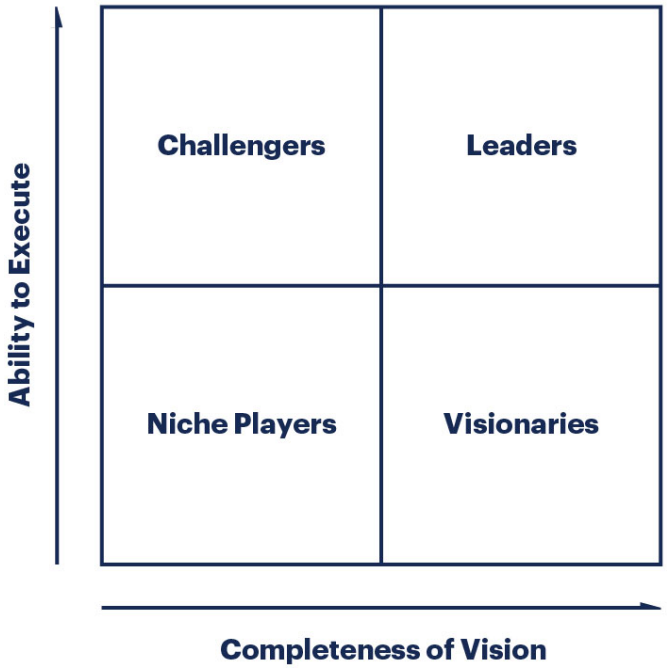
Σχήμα 22 – Τα πάντα είναι στην ευθύνη του πελάτη

Το μοντέλο που παρουσιάζεται στο Σχήμα 21 –Υλικό και host os υλοποιούνται σε εγκαταστάσεις εκτός ευθύνης πελάτη μπορεί να αφορά και Open Source Cloud Platforms αλλά και Proprietary Cloud Platforms. Αντίθετα, το μοντέλο στο Σχήμα 22 – Τα πάντα είναι στην ευθύνη του πελάτη μπορεί να υλοποιηθεί μόνο στη περίπτωση του Open Source Cloud Platform.

7.1 Μερίδιο αγοράς και τάσεις

Το Μάιο του 2018, η εταιρεία Gartner (https://www.gartner.com) δημοσίευσε το Magic Quadrant for Cloud Infrastructure as a Service, Worldwide [41], παρουσιάζοντας τη θέση των παρόχων υπηρεσιών cloud σε σχέση με την αγορά.

Το Gartner Magic Quadrant είναι ένας τρόπος απεικόνισης των αποτελεσμάτων έρευνας μίας συγκεκριμένης αγοράς. Ειδικότερα, οι εταιρείες τις οποίες αφορά τοποθετούνται σε ένα πλέγμα 4 τετραγώνων.



Σχήμα 23 - Γενική μορφή Magic Quadrant

Οι εταιρείες στον οριζόντιο άξονα αξιολογούνται ως προς τους στόχους τους και το όραμα τους και κατά πόσο το επιτυγχάνουν. Στον κατακόρυφο άξονα τοποθετούνται με βάση την ικανότητα υλοποίησης των στόχων τους και υποστήριξης του έργου τους.

Σχήμα 24 - Magic Quadrant for Cloud Infrastructure as a Service, Worldwide [41]

Σύμφωνα με τη Gartner (Σχήμα 24 - Magic Quadrant for Cloud Infrastructure as a Service, Worldwide [41]), οι πρώτες 2 θέσεις ανήκουν στο Amazon AWS και το Microsoft Azure

ακολουθώντας με σημαντική διαφορά στην τρίτη θέση το Google Cloud Platform. Ανοδική πορεία ακολουθεί το Alibaba Cloud με παραδοσιακά μεγάλες εταιρείες όπως IBM και Oracle να υπολείπονται κυρίως σε θέματα υλοποίησης.

7.2 Open Source Cloud Platforms

Παρόλο που δεν υπάρχει επίσημη αποτύπωση από τη Gartner για το μερίδιο αγοράς που κατέχουν οι ανοιχτές πλατφόρμες IaaS cloud computing, γίνεται μία σύντομη επισκόπηση στις πιο γνωστές από αυτές.

7.2.1 OpenStack

Πρόκειται για την πιο διαδεδομένη πλατφόρμα δημιουργία και διαχείρισης IaaS cloud – private και public. Ανήκει στην κατηγορία του λογισμικού ΕΛΛ/ΛΑΚ. Η άδεια χρήσης του είναι η Apache License, version 2.0 (ASLv2) - http://www.apache.org/licenses/LICENSE-2.0. Απευθύνεται σε παρόχους υπηρεσιών, κυβερνητικές υποδομές, ακαδημαϊκά ιδρύματα αλλά και επιχειρήσεις μεγάλου και μεσαίου μεγέθους, ανεξαρτήτως τομέα δραστηριότητας. Χρησιμοποιεί δομή αρθρωτή, υπό την έννοια ότι αποτελείται από πολλά διαφορετικά υποέργα. Το κάθε υποέργο που ενσωματώνεται σε μία υπάρχουσα υλοποίηση προσθέτοντας σε αυτήν νέες δυνατότητες. Τα υποέργα χωρίζονται ανάλογα με την υπηρεσία που παρέχουν στις παρακάτω κατηγορίες

- Compute
- Bare Metal
- Storage
- Networking
- Shared Services
- Orchestration
- Workload Provisioning
- Application Lifecycle
- API Proxies
- Web Frontend

Τα πιο δημοφιλή αλλά και ταυτόχρονα τα ελάχιστα απαραίτητα projects είναι τα

- Nova (compute)
- Neutron (networking)
- Swift (Object Storage)
- Glance (Imaging Service)
- KeyStone (Identity Service)
- Cinder (Block Storage)
- Horizon (DashBoard)

Επίσης παρέχονται έτοιμα αρχεία ρυθμίσεων και διαμορφώσεις για τις πιο συνήθεις χρήσεις του OpenStack

- Web Applications
- Big Data
- eCommerce
- Video Processing and Content Delivery
- High Throughput Computing
- Container Optimized
- Web Hosting
- Public Cloud
- Compute Starter Kit

- DBaaS

Για κάθε επιπλέον απαίτηση αλλά και για περισσότερες έτοιμες λύσεις υπάρχει το MarketPlace (https://www.openstack.org/marketplace/). Τέλος το έργο υποστηρίζεται από μεγάλες εταιρείες του χώρου της Πληροφορικής αλλά και από πολύ μεγάλη κοινότητα χρηστών.

7.2.2 Apache CloudStack

Η δεύτερη σε αποδοχή λύση για τη δημιουργία και διαχείριση IaaS cloud. Υποστηρίζει τη δημιουργία private και public cloud. Είναι ανοιχτού κώδικα και η άδεια χρήσης είναι η Apache License, version 2.0 (ASLv2). Απευθύνεται σε παρόχους υπηρεσιών και μεγάλους οργανισμούς. Η δομή του, σε αντίθεση με το OpenStack είναι ενιαία. Όλες οι λειτουργίες υλοποιούνται μέσα στο ίδιο το project. Αυτό σημαίνει πως από τη στιγμή της επιτυχούς εγκατάστασής του είναι διαθέσιμες όλες οι επιλογές και δυνατότητες. Το έργο υποστηρίζεται από μικρότερη κοινότητα χρηστών σε σχέση με το OpenStack αλλά υποστηρίζεται από το Apache Software Foundation (http://apache.org/), έναν από τα μεγαλύτερους οργανισμούς στον τομέα του open source.

7.2.3 OpenNebula

Μία ακόμα open source λύση για τη δημιουργία και διαχείριση private IaaS cloud. Διατίθεται κάτω από την άδεια χρήσης η Apache License, version 2.0 (ASLv2). Θεωρείται απλούστερο από τα OpenStack και CloudStack ως προς την εγκατάσταση, τη ρύθμιση και τη συντήρησή του. Υπολείπεται σε διαθέσιμες επιλογές ως προς τη χρήση του αλλά μπορεί να καλύψει τις απαιτήσεις ενός μεγάλου εταιρικού φορέα ή παρόχου υπηρεσιών. Η αρχιτεκτονική του είναι μερικώς αρθρωτή υποστηρίζοντας πρόσθετα (addons) που επεκτείνουν τη λειτουργικότητά του. Βασικό πλεονέκτημα του OpenNebula είναι η ύπαρξη εμπορικής εταιρείας πίσω από το project. Αυτό σημαίνει – εκτός από την υποστήριξη της κοινότητας – εμπορική υποστήριξη από τους προγραμματιστές του ίδιου του έργου.

7.2.4 oVirt

Open source λύση, με ενεργή κοινότητα από πίσω, παρέχει distributed virtualization για δημιουργία και διαχείριση μίας ολόκληρης επιχειρησιακής υποδομής. Το μοντέλο της παρεχόμενης υπηρεσίας είναι το IaaS. Κάνει χρήση του KVM hypervisor, που σημαίνει πως ως host os μπορεί να χρησιμοποιηθεί μόνο το linux. Διατίθεται κάτω από την άδεια χρήσης η Apache License, version 2.0 (ASLv2). Μελετώντας μερικά από τα User Case Studies γίνεται αντιληπτό πως πρόκειται για μία ολοκληρωμένη πρόταση διαχείρισης υποδομής ακόμα και σε μεγάλης κλίμακας οργανισμούς και πανεπιστήμια.

7.3 Proprietary Cloud Platforms

Στη συνέχεια γίνεται μία σύντομη αναφορά για τις εταιρείες των οποίων οι υπηρεσίες αποτυπώνονται στο Gartner Magic Quadrant [41] προκειμένου ο αναγνώστης να αποκτήσει μία συνολική εικόνα.

7.3.1 Amazon WS

Ένας από τους πρωτοπόρους του cloud computing με σημαντικό μερίδιο στην αγορά είναι η Amazon. Η εταιρεία προσφέρει μεγάλο αριθμό προϊόντων, με το καθένα να μπορεί να λειτουργήσει ως άρθρωμα σε μία ευρύτερη υποδομή. Τα προϊόντα αφορούν υπηρεσίες όπως

- Compute/Virtual servers
- Scalable storage
- High performance RDBMS
- Managed NoSQL database
- Code execution
- Resource Isolation
- Identity management
- Networking

Τα προϊόντα της amazon δίνουν στους πελάτες της τη δυνατότητα να υποστηρίξουν τις σύγχρονες τάσεις της τεχνολογίας. Ενδεικτικά αναφέρονται μερικές κατηγορίες λύσεων που δίνει η εταιρεία :

- Blockchain
- GameTech
- Internet of Things
- Machine Learning
- Migration & Transfer
- Mobile
- Content Delivery
- Robotics
- Satellite
- Security

Όλα τα προϊόντα μπορούν να συνεργαστούν μεταξύ τους και αποτελούν διαχειρίσιμα συστατικά μέσα από πίνακα ελέγχου. Πλήρης κατάλογος των λύσεων για κάθε περίπτωση υπάρχει στο αντίστοιχο Marketplace (https://aws.amazon.com/marketplace/) της εταιρείας.

7.3.2 VMware vCloud Suite

Το Vmware vCloud Suite είναι ένα πλήρες πακέτο εμπορικού και κλειστού λογισμικού της εταιρείας VMWare. Το πακέτο περιλαμβάνει το hypervisor λογισμικό της εταιρείας, το vSphere Plus και το λογισμικό διαχείρισης cloud vRealize Suite. Το μοντέλο ανάπτυξης cloud που υποστηρίζεται είναι το υβριδικό, υποστηρίζοντας όμως και αποκλειστικά private clouds. Οι υπηρεσίες και οι δυνατότητες που παρέχει υποστηρίζουν τη διαχείριση Software Defined Data Center (SDDC) καθώς και αυτοματισμούς απαραίτητους για την ανάπτυξη και την εκτέλεση εφαρμογών πάνω στο cloud. Η αδειοδότηση χωρίζεται σε πακέτα (Standard, Advanced, Enterprise και custom) όπου κάθε περίπτωση αφορά από απλή SaaS υλοποίηση

μέχρι IaaS η πιο σύνθετη. Μερικές από τις κατηγορίες των προϊόντων του vCloud Suite είναι :

- SDDC Platform
- NetWokring
- Storage & Availability
- Internet of Things
- Digital Workspace
- Desktop and Application
- Virtualization

Αξίζει να σημειωθεί πως το προϊόν vSphere hypervisor, που ανήκει στην κατηγορία bare-metal, διατίθεται δωρεάν.

7.3.3 Google Cloud Platform

Η cloud πλατφόρμα από τη Google, η οποία συμπεριλαμβάνει και το Google G Suite, κάνει χρήση της υλικής υποδομής που τρέχουν τα δημοφιλή προϊόντα της ίδιας της εταιρείας (Google Search, YouTube) . Οι υπηρεσίες εκτείνονται από το IaaS μοντέλο μέχρι και το SaaS μοντέλο προσφέροντας τη δυνατότητα Serverless computing . Οι λύσεις που παρέχονται δύναται να καλύψουν από τις ανάγκες μίας πρώιμης startup εταιρείας μέχρι αυτές ενός μεγάλου οργανισμού. Τα προϊόντα της μπορούν να κατηγοριοποιηθούν με βάση τον τομέα της επιχείρησης :

- Εκπαίδευση
- Ενέργεια
- Οικονομικές υπηρεσίες
- Gaming
- Ηλεκτρονική διακυβέρνηση
- Επιστήμες υγείας
- Κοινωνικές επιστήμες
- Μέσα ενημέρωσης

Πιο συγκεκριμένα όμως μερικές (συνολικά είναι πάνω από 100) από τις τεχνολογίες στις οποίες η Google δίνει μέσω cloud δυνατότητα εκμετάλλευσης της υποδομής της είναι :

- AI and Machine Learning
- Διαχείριση APIs
- Compute
- Data Analytics
- Βάσεις δεδομένων
- Containers
- Internet of Things
- Migration
- Networking
- Security
- Storage

Και σε αυτή την περίπτωση η υποστήριξη είναι συνεχής ενώ προβλέπεται και η παροχή εκπαίδευσης από την εταιρεία.

7.3.4 Microsoft Azure

Η πρόταση της Microsoft στο χώρο του IaaS cloud computing είναι το Microsoft Azure. Πρόκειται για ένα συνεχώς ανανεώσιμο σύνολο από υπηρεσίες και προϊόντα με

δυνατότητες δημιουργίας και διαμοιρασμού υλικού και λογισμικού. Περιλαμβάνει έτοιμες λύσεις αλλά και custom στα μέτρα του κάθε πελάτη. Υποστηρίζει και τις ανοιχτές πλατφόρμες όπως linux, πέρα από τα προϊόντα της Microsoft. Μερικές από τις κατηγορίες για τις οποίες παρέχει λύσεις μέσω cloud η Microsoft είναι :

- AI & Machine Learning
- Analytics
- Containers
- Databases
- Developer tools
- Identity Management
- Internet of Things
- Management
- Media
- Networking
- Security
- Storage

Όπως στην περίπτωση της Google, έτσι και η Microsoft προσφέρει περισσότερα από 100 εργαλεία που μπορεί να χρησιμοποιήσει και να συνδυάζει κάποιος στο Azure cloud καλύπτοντας ολόκληρο το φάσμα της δραστηριότητας οργανισμών και επιχειρήσεων κάθε μεγέθους.

7.3.5 Alibaba Cloud

Μία προσπάθεια στο χώρο του IaaS cloud computing από τη νεότερη από τις παραπάνω εταιρείες, η Alibaba, έχει κερδίσει σοβαρό μερίδιο της αγοράς, διεκδικώντας τον τίτλο του νέου μεγάλου “παίχτη” στο χώρο. Έχει κερδίσει την εμπιστοσύνη μεγάλων εταιρειών-πελατών και όχι μόνο. Χαρακτηριστικό παράδειγμα είναι η συμφωνία για κάλυψη των αναγκών των Ολυμπιακών Αγώνων μέχρι και το 2028 ύστερα από συμφωνία με τη Διεθνή Ολυμπιακή Επιτροπή. Η παραπάνω συμφωνία αποδεικνύει τη δυνατότητα κάλυψης μεγάλου όγκου εργασιών αλλά και την ύπαρξη της αντίστοιχης τεχνογνωσίας προκειμένου να ανταποκριθεί η εταιρεία στις σύγχρονες απαιτήσεις. Τα προϊόντα της κατηγοριοποιούνται σε

- Computing
- Elasticity
- Application Servers
- Storage
- Networking
- Databases
- Security
- Monitoring
- Domains and Websites
- Analytics
- Big Data
- Middleware
- Cloud Communication
- Internet of Things

7.3.6 Oracle Cloud

Η Oracle διατηρώντας ένα σταθερό τμήμα της αγοράς cloud υπηρεσιών, βρίσκεται αυτή τη στιγμή (2019) στη δεύτερη γενιά εργαλείων cloud. Προσφέρει εικονική υποδομή η οποία ξεπερνά σε αποδοτικότητα και υπηρεσίες τα σύγχρονα datacenters. Παράλληλα παρέχει

όλες τις υπηρεσίες των public clouds όπως scaling, elasticity κλπ. Οι βασικές κατηγορίες των υπηρεσιών συνοψίζονται σε :

- Compute
- Networking
- Storage
- Database
- Containers
- Security
- Connectivity
- Optimization
- Edge Services

7.3.7 IBM Cloud

Η πλατφόρμα που παρέχει η IBM υλοποιεί public, private και hybrid μοντέλα cloud. Οι δυνατότητες των υπηρεσιών είναι τέτοιες που μπορεί να καλύψουν από τις ανάγκες μίας μικρής επιχείρησης έως και αυτές ενός μεγάλου οργανισμού. Μερικά use cases αφορούν σε

- Websites & Web Application
- Backup & Restore
- SaaS Migration
- DevOps
- API Handling
- GPU Computing

Συνολικά οι προσφερόμενες λύσεις της IBM στο cloud ξεπερνούν τις 170 με κυριότερες κατηγορίες τις :

- Compute
- Network
- Storage
- Management
- Security
- Databases
- Analytics
- AI
- IoT
- Mobile
- Developer Tools
- BlockChain
- Integration & APIs
- Migration
- VMWare

Ωστόσο, 3 γεγονότα είναι αξιοσημείωτα και δείχνουν την προσπάθεια της εταιρείας να καθιερωθεί στο χώρο του cloud αποκτώντας πρωταγωνιστικό ρόλο.

a. Η IBM σύμφωνα με το OpenStack (https://www.openstack.org/foundation/companies/) ανήκει στους Corporate Sponsors του project. Ταυτόχρονα η Red Hat ανήκει στα Platinum Members του OpenStack.
b. Οι συντάκτες και οι contributors των [3], [39] που παρουσιάζουν το πρότυπο CADF είναι μηχανικοί της IBM.
c. Η εξαγορά της Red Hat από την IBM – όπου και η ίδια η Red Hat είναι πάροχος υπηρεσιών cloud.

Τα παραπάνω αποτελούν σημεία της προσπάθειας της IBM από απλός παίχτης στην αγορά υπηρεσιών cloud να γίνει πρωταγωνιστής. Ταυτόχρονα όμως αποκαλύπτουν και την ευμεταβλητότητα του τοπίου των IaaS cloud υπηρεσιών.

Κεφάλαιο 8
OpenStack/DevStack Use Case

Σκοπός είναι η εξέταση των αρχείων καταγραφής που παράγονται κατά την εκτέλεση σεναρίων με συγκεκριμένες ενέργειες. Το OpenStack είναι το έργο για το οποίο αναπτύχθηκε το μοντέλο CADF και είναι το μοναδικό μέχρι σήμερα που το υποστηρίζει επίσημα ως προεπιλεγμένη μορφή εξόδου των αρχείων καταγραφής.

8.1 Δομή και τρόπος λειτουργίας

Ένα βασικό συστατικό του OpenStack αποτελεί το project Keystone, που παρέχει λειτουργίες διαχείρισης ταυτότητας. Οι υπηρεσίες του καθίστανται διαθέσιμες με τη βοήθεια API endpoints.Οι υπηρεσίες κατηγοριοποιούνται ως εξής :

- Identity
- Resource
- Assignment
- Token
- Catalog

Η υπηρεσίες Identity ασχολούνται με τον έλεγχο των διαπιστευτηρίων, τις πληροφορίες σχετικά με τους χρήστες (users) και τις ομάδες των χρηστών (groups). Πολλές φορές ολόκληρο το project Keystone αναφέρεται και ως Identity Service. Οι Resource υπηρεσίες αφορούν στα Projects και στα Domains. Το Project αποτελεί μία βασική δομική μονάδα στην οποία πρέπει υποχρεωτικά να ανήκουν οι πόροι. Δε μπορεί δηλαδή π.χ. μία εικονική μηχανή να μην ανήκει σε ένα Project. Σε ακόμα υψηλότερο επίπεδο βρίσκεται το Domain, μία οργανωτική μονάδα. Ένα Domain μπορεί να περιέχει πολλά Projects. Οι Assignment υπηρεσίες διαχειρίζονται τα roles και role assignments. Το role καθορίζει το επίπεδο πρόσβασης και τα δικαιώματα που μπορεί να κατέχει ένας χρήστης. Το role assignment είναι ένα τρίπτυχο που ορίζει επακριβώς ποιος χρήστης, έχει ποιο role πάνω σε ποιο resource.

Δομικό Στοιχείο	Μοναδικότητα

Domain	μοναδικό καθολικά
Project	μοναδικό μόνο μέσα στο domain, όχι καθολικά
User	μοναδικό μόνο μέσα στο domain, όχι καθολικά
Group	μοναδικό μόνο μέσα στο domain, όχι καθολικά

Πίνακας 8 - Δοκιμά στοιχεία OpenStack

Μετά την επικύρωση των διαπιστευτηρίων ενός χρήστη, για συγκεκριμένο χρονικό διάστημα δεν ζητείται η επανεισαγωγή τους. Για κάθε ενέργεια όμως που εκτελεί ο συγκεκριμένος είναι απαραίτητη η επιβεβαίωση της ταυτότητας του. Αυτό επιτυγχάνεται με τις υπηρεσίες Token.

Τέλος, οι υπηρεσίες καταλόγου χρησιμοποιούνται για την ανακάλυψη και καταχώρηση των endpoints. (https://docs.openstack.org/keystone/latest/getting-started/architecture.html)

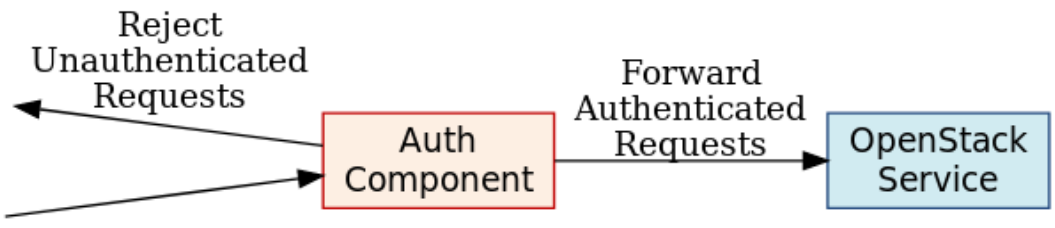
8.1.1 Auditing και Logging

Στο project Keystone ενσωματώνεται το υποέργο “Middleware for OpenStack Identity”. Το middleware παρέχει υπηρεσίες αυθεντικοποίησης και εξουσιοδότησης στις υπόλοιπες υπηρεσίες του OpenStack, εκτός από το Keystone. Αλληλεπιδρά με το Identity API και επιβεβαιώνει την ταυτότητα του χρήστη με τη χρήση bearer tokens[13]. Ένα πολύ σημαντικό χαρακτηριστικό του middleware είναι η δυνατότητα δημιουργίας συμβάντων προς καταγραφή για API requests άλλων υπηρεσιών του OpenStack προς το Keystone.

Κάθε συστατικό του OpenStack/DevStack ρυθμίζεται ξεχωριστά. Δηλαδή, το Keystone μπορεί να παράγει logs που να έχουν τη μορφή CADF ενώ το Nova (compute) να έχει διαφορετική μορφή.

8.1.2 Αρχιτεκτονική του middleware

Κάθε αίτηση για υπηρεσία του OpenStack είτε μέσω API είτε μέσω client είναι απαραίτητο να προέρχεται από διαπιστευμένο και εξουσιοδοτημένο χρήστη. Ο έλεγχος γίνεται από την υπηρεσία διαχείρισης ταυτότητας (Keystone). Αν η αίτηση είναι έγκυρη τότε προωθείται προς την αντίστοιχη υπηρεσία (πχ nova, neutron κλπ.) προς διεκπεραίωση. Σε αντίθετη περίπτωση το αίτημα για υπηρεσία απορρίπτεται.



Σχήμα 25 - Authentication Component in OpenStack[42]

[13] Δίνεται πρόσβαση σε αυτόν που “φέρει” το token

8.1.3 Pipelines και WSGI Middleware

Κάθε αίτηση για οποιαδήποτε ενέργεια στο OpenStack προκειμένου να εξεταστεί και να διεκπεραιωθεί μπαίνει σε μία διαδικασία σειριακής επεξεργασίας που ονομάζεται pipeline. Σε ένα pipeline η έξοδος της κάθε διεργασίας αποτελεί είσοδο της επόμενης.

Σε περιβάλλον web, παρουσιάζεται συχνά η ανάγκη ο web server να προωθήσει ένα request σε άλλες web εφαρμογές. Στη γλώσσα προγραμματισμού Python αυτό υλοποιείται με τη χρήση του WSGI Middleware. Το middleware είναι WSGI Application το οποίο χειρίζεται εισερχόμενες αιτήσεις αναθέτοντας τις σε άλλα WSGI Applications

https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface

Το auditing middleware του Keystone μπορεί να ενεργοποιηθεί για οποιοδήποτε project του OpenStack. Στην πραγματικότητα το WSGI middleware το οποίο χειρίζεται το pipeline του web request του κάθε project μεταβάλλεται έτσι ώστε να παρεμβάλλεται στη διαδικασία και το audit middleware

8.2 Εγκατάσταση και δοκιμή

Το σύστημα που χρησιμοποιήθηκε για τις δοκιμές έχει τα εξής χαρακτηριστικά:

- Intel® Pentium(R) CPU 2020M @ 2.40 GHz, dual-core
- 8GB RAM
- HDD 250 GB SATA
- Ubuntu 18.04 LTS (Bionic Beaver) 64-bit
- Kernel Linux 4.15.0-43-generic x86_64
- MATE 1.20.1

Στις οδηγίες εγκατάστασης του OpenStack (https://docs.openstack.org/install-guide/overview.html) αναφέρεται πως απαιτούνται τουλάχιστον 2 hosts ως ελάχιστη δυνατή εγκατάσταση – ένας host θα έχει το ρόλο του Controller node και ένας του Compute node. Επίσης κάθε κόμβος πρέπει να διαθέτει τουλάχιστον 8 GB RAM και τουλάχιστον 2 επεξεργαστές και 2 κάρτες δικτύου. Αν χρησιμοποιηθεί μόνο ένας κόμβος τότε οι απαιτήσεις αθροίζονται με αποτέλεσμα να είναι αρκετά υψηλές για ένα οικιακό σύστημα. Επίσης, οι αλλαγές που θα προκύψουν σε ένα σύστημα μετά από πλήρη εγκατάσταση του OpenStack το καθιστούν μη χρηστικό για συνηθισμένες εργασίες. Το πρόβλημα της αδυναμίας εγκατάστασης θα μπορούσε να αντιμετωπιστεί με λύσεις όπως :

- Χρήση Virtualization
- Χρήση Containers

Και στις 2 περιπτώσεις απαιτούνται από το σύστημα τουλάχιστον οι ίδιοι πόροι με αυτούς των εικονικών μηχανών. Δηλαδή, δε μπορεί ένας host με 8 GB RAM να δώσει σε εικονικές μηχανές 16 GB RAM.

Η λύση που επιλέχθηκε είναι το DevStack (https://docs.openstack.org/devstack/latest/). Το DevStack είναι σύνολο από scripts τα οποία εγκαθιστούν ένα πλήρες περιβάλλον OpenStack απευθείας από το κύριο αποθετήριο χρησιμοποιώντας κώδικα αντί για έτοιμα πακέτα. Τα scripts εκτός από την εγκατάσταση των συστατικών αναλαμβάνουν και τη ρύθμισή τους ώστε να παρέχεται η βασική λειτουργικότητα. Χρησιμοποιείται ως περιβάλλον ανάπτυξης και αποσφαλμάτωσης του OpenStack. Οι απαιτήσεις σε υλικό είναι σαφώς μικρότερες, καθώς οι ρυθμίσεις με βάσει τις οποίες εγκαθίσταται περιορίζουν τις δυνατότητες ως προς το scaling, το elasticity το πλήθος και είδος των υποστηριζόμενων εικονικών πόρων. Η δυνατότητα χρήσης περισσοτέρων από ένα domains είναι απενεργοποιημένη. Για αυτούς τους λόγους δεν αποτελεί καλή επιλογή για συστήματα production, αλλά είναι ιδανική επιλογή για το ζήτημα που πραγματεύεται η παρούσα διατριβή.

8.2.1 Εγκατάσταση DevStack

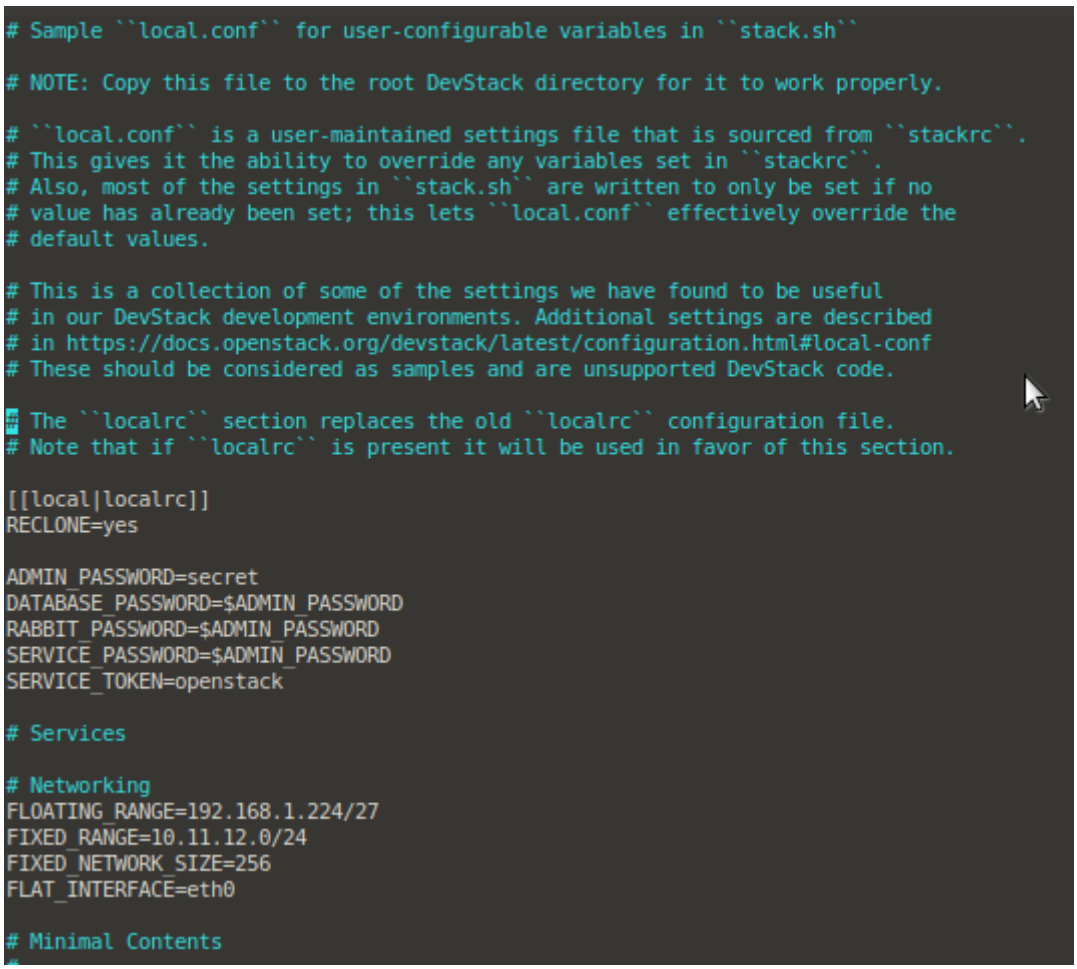
Για λόγους ευκολίας και αυτοματισμού δημιουργήθηκε το αποθετήριο *os_helpers* στο GitHub (https://github.com/ndalezios/os_helpers) το οποίο περιλαμβάνει scripts για διάφορα στάδια της εγκατάστασης και ρύθμισης του συστήματος.

Πριν ξεκινήσει η εγκατάσταση εκτελείται το script *0_user_add.sh* που δημιουργεί νέο χρήστη με όνομα *stack*, δικαίωμα sudo και κάνει login με το νέο χρήστη

```
#!/bin/bash
useradd –s /bin/bash –d /opt/stack –m stack
echo “stack ALL=(ALL) NOPASSWD: ALL” | sudo tee /etc/sudoers.d/stack
#now user is stack
su – stack
```

Script 1 – 0_user_add.sh

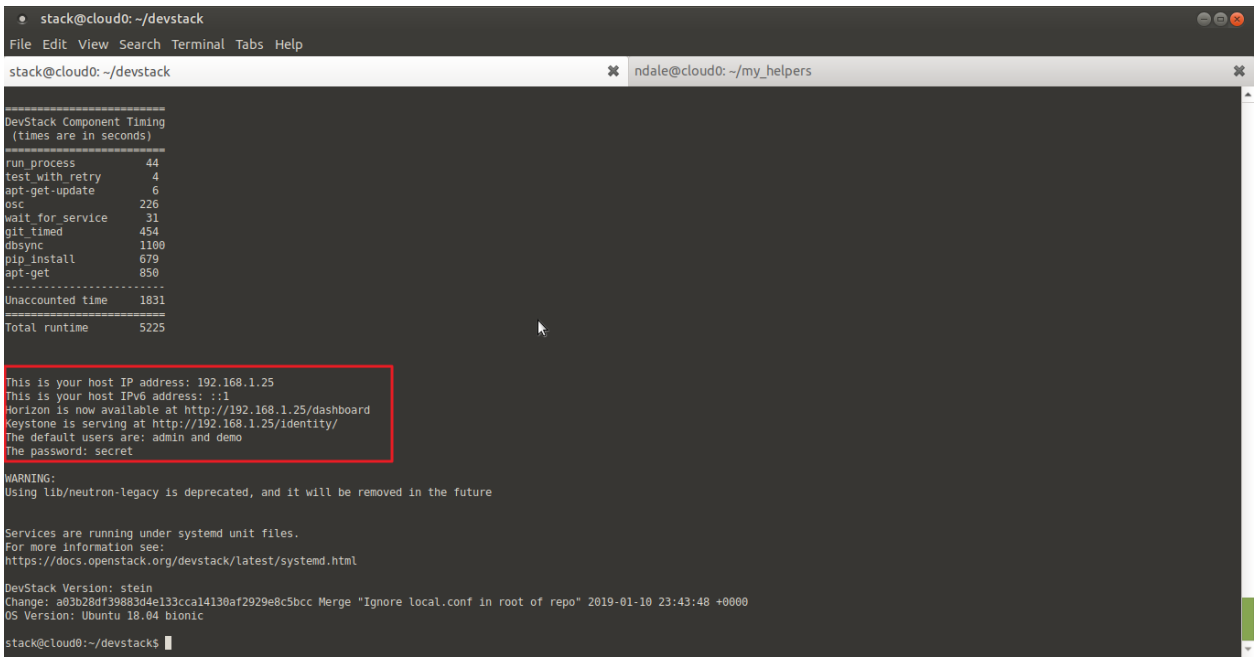
Ακολουθεί η λήψη του πηγαίου κώδικα του DevStack μέσω του αποθετηρίου του (https://git.openstack.org/openstack-dev/devstack). Πριν την εκκίνηση της εγκατάστασης απαιτείται η δημιουργία ενός αρχείου *local.conf* με περιεχόμενα τα συνθηματικά και κάποιες βασικές ρυθμίσεις δικτύου. Το αρχείο που χρησιμοποιήθηκε υπάρχει στο αποθετήριο *os_helpers* και είναι το default sample με τις ακόλουθες αλλαγές



Στιγμιότυπο 1 - Αρχείο βασικών ρυθμίσεων της διαδικασίας εγκατάστασης του DevStack

Στο local.conf εκτός από το συνθηματικό "*secret*" που ορίζεται για όλες τις υπηρεσίες, η ρύθμιση "RECLONE=yes" υποχρεώνει το script εγκατάστασης κάθε φορά να κάνει clone ξανά το αποθετήριο. Ακόμα ορίζονται οι διαθέσιμες διευθύνσεις για private δίκτυο (FIXED_RANGE) και για public δίκτυο (FLOATING).

Η διαδικασία της εγκατάστασης μπορεί τώρα να ξεκινήσει, ωστόσο απαιτεί αρκετό χρόνο.



Στιγμιότυπο 2 - Ολοκλήρωση της διαδικασίας εγκατάστασης του DevStack

Μετά την επιτυχή ολοκλήρωση της εγκατάστασης, ενημερώνεται ο χρήστης για τις βασικές παραμέτρους όπως IP address, διεύθυνση Horizon (Dashboard), διεύθυνση Keystone (Identity), default χρήστες και συνθηματικό. Στο συγκεκριμένο σύστημα η εγκατάσταση του DevStack ολοκληρώθηκε σε 5225 δευτερόλεπτα (περίπου 87 λεπτά).

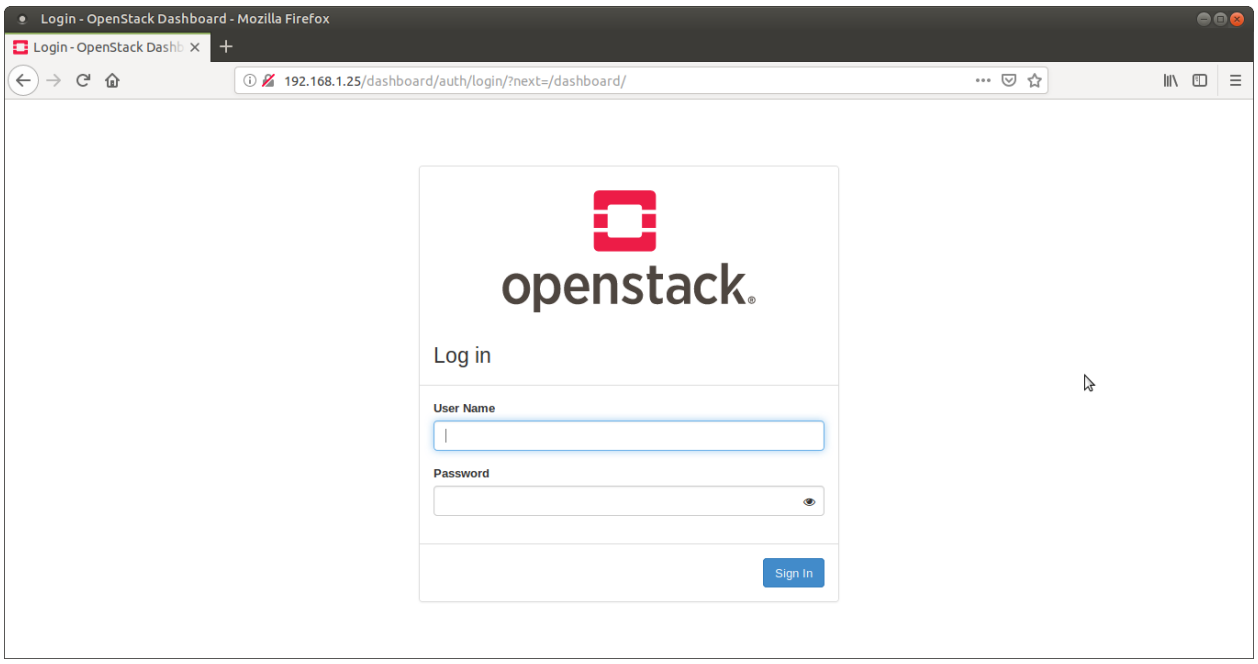
Στην περίπτωση που η εγκατάσταση αποτύχει για κάποιο λόγο, οι αλλαγές που έχουν γίνει στο σύστημα αφαιρούνται από τα scripts του DevStack *unstack.sh* και *clean.sh*. Στο σύστημα όμως συνεχίζουν να μένουν αλλαγές που θα εμποδίζουν επόμενη εγκατάσταση. Για το λόγο αυτό καλό είναι να μην καλούνται μεμονωμένα (*unstack.sh* και *clean.sh*) αλλά να γίνεται χρήση του *100_remove_devstack.sh* script από το αποθετήριο *os_helpers*.

```
#!/bin/bash
/opt/stack/devstack/unstack.sh
/opt/stack/devstack/clean.sh
rm -rf /opt/stack/devstack
rm -rf /usr/local/bin/
cd /usr/local
rm -rf lib/python2.7/dist-packages/* site_ruby/* bin/*
```

Script 2 - 100_remove_devstack.sh

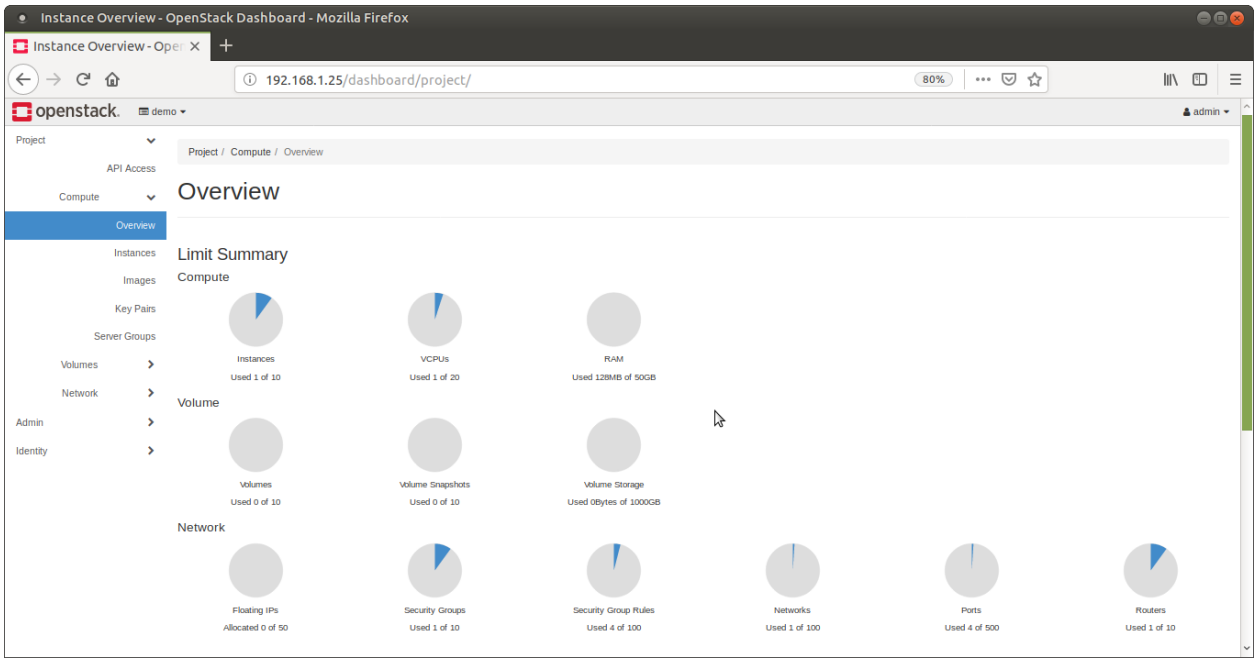
Το *100_remove_devstack.sh* εκτελεί τα *unstack.sh* και *clean.sh* και στη συνέχεια διαγράφει προσωρινά αρχεία, εκτελέσιμα αρχεία, αρχεία κώδικα του DevStack και πακέτα και ρυθμίσεις python και ruby που έγιναν από το script εγκατάστασης.

Εφόσον η εγκατάσταση ολοκληρώθηκε με επιτυχία, στη διεύθυνση που εμφανίζει η σύνοψη της εγκατάστασης θα πρέπει να εμφανίζεται το login panel του Horizon (dashboard).

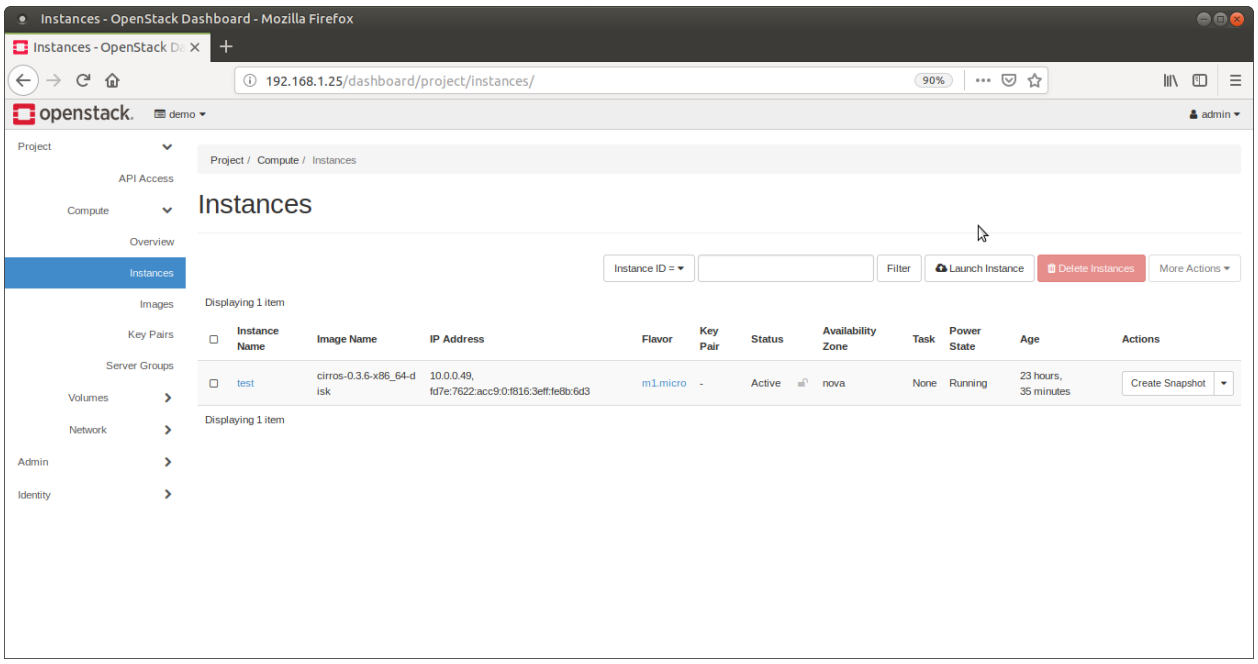


Στιγμιότυπο 3 - Οθόνη εισόδου στο dashboard του DevStack

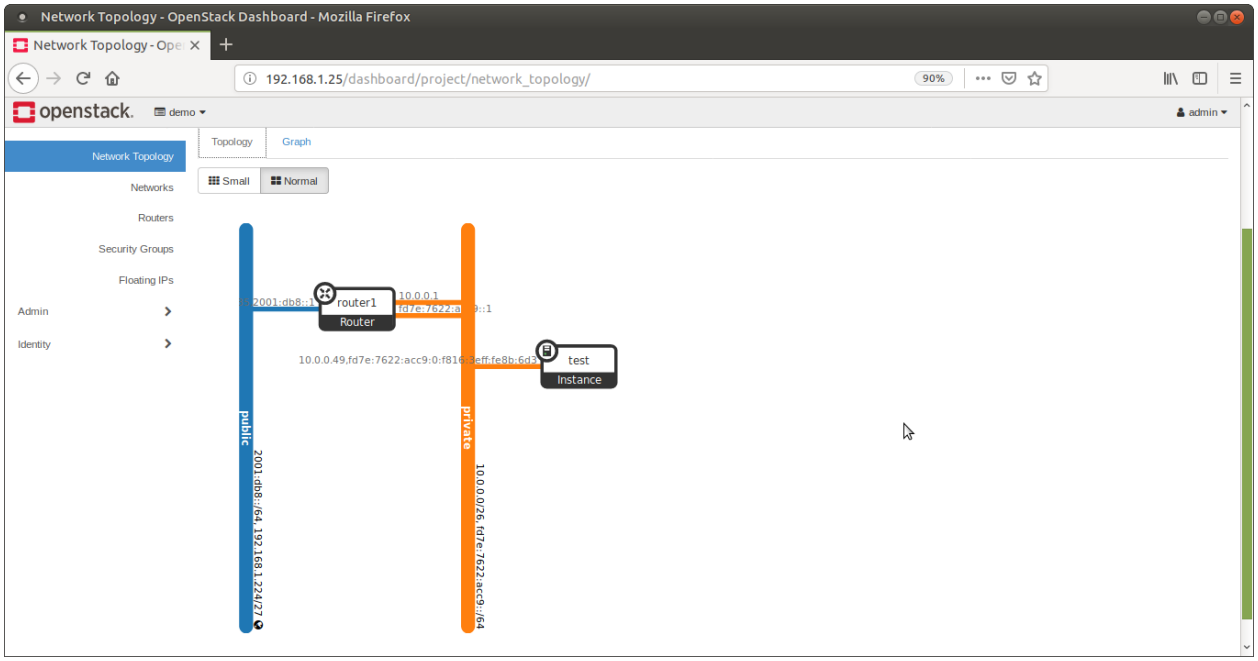
Μετά την διαδικασία αυθεντικοποίησης εμφανίζεται η σύνοψη των εικονικών πόρων και της χρήσης τους.



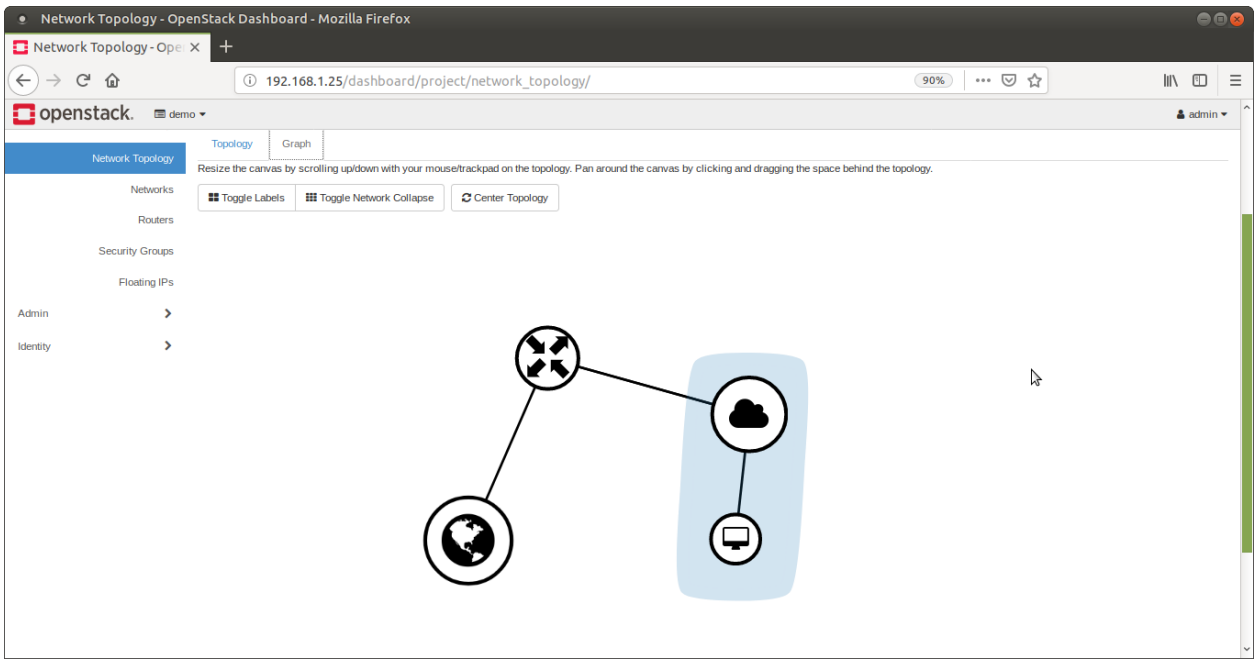
Στιγμιότυπο 4 - Σύνοψη εικονικών πόρων



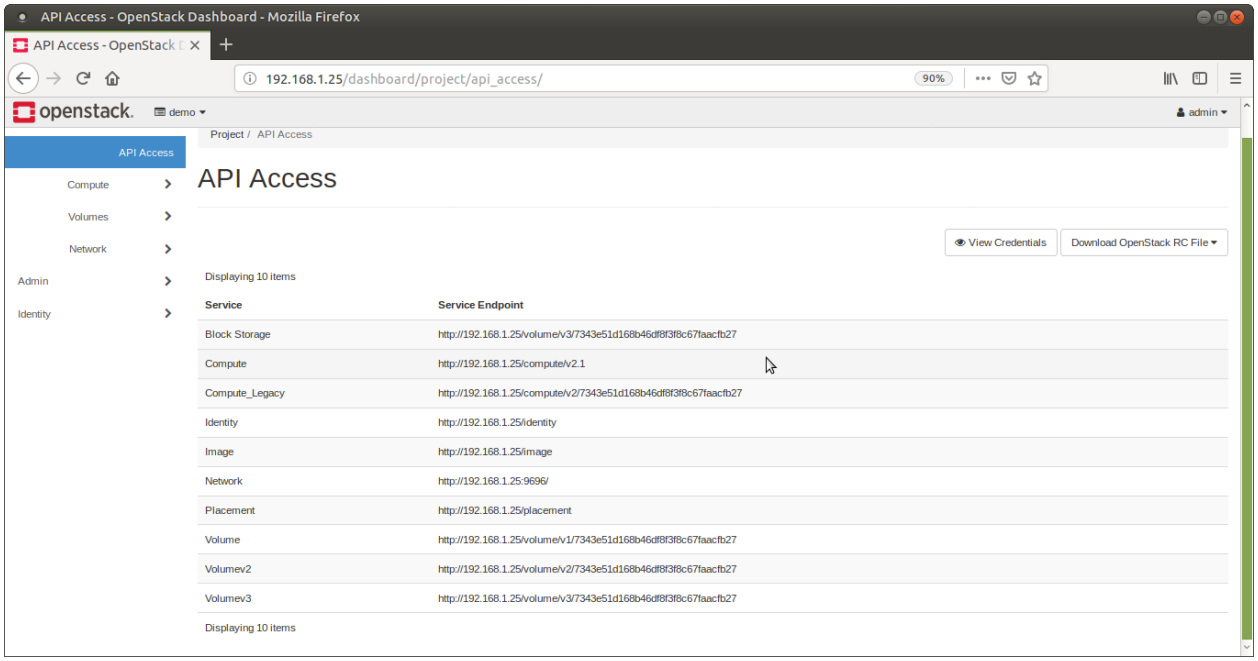
Στιγμιότυπο 5 - Εικονική μηχανή "test" σε κατάσταση "running"



Στιγμιότυπο 6 – Τοπολογία ιδιωτικού δίκτυο, εικονικού router και δημόσιου δικτύου



Στιγμιότυπο 7 - Γραφική απεικόνιση του δικτύου



Στιγμιότυπο 8 - API endpoints

8.2.2 Βασικές Ρυθμίσεις

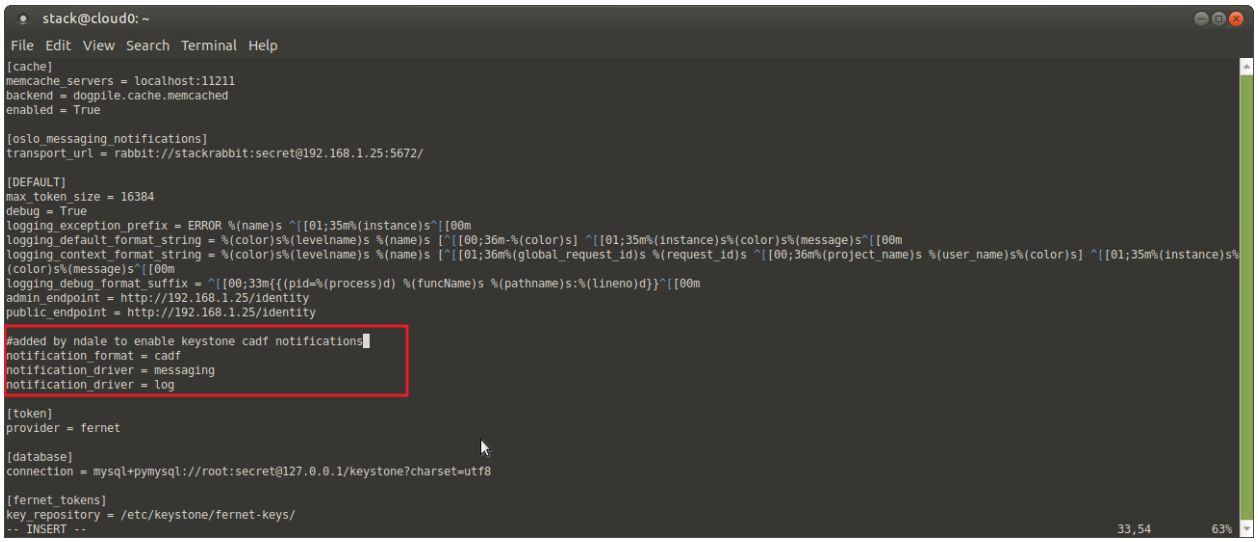
Για την εμφάνιση και την επεξεργασία των audit events συχνά χρησιμοποιείται κάποιο plugin που έχει το ρόλο του message broker και "διοχετεύει" τις ειδοποιήσεις σε ουρές (queues). Οι ουρές αυτές είναι προσβάσιμες μέσω web interface. Στο DevStack είναι προεγκατεστημένο το RabbitMQ (https://www.rabbitmq.com/). Για την ενεργοποίηση, τη ρύθμιση και τον ορισμό συνθηματικών χρησιμοποιείται από το αποθετήριο *os_helpers* το script *50_configure_rabbitmq.sh*.

```
#!/bin/bash
```

```
rabbitmq-plugins list
rabbitmq-plugins enable rabbitmq_management
service rabbitmq-server restart
rabbitmqctl change_password stackrabbit guest
rabbitmqctl set_user_tags stackrabbit administrator
```

Script 3 - 50_configure_rabbitmq.sh

Στη συνέχεια, πραγματοποιείται αλλαγή των ρυθμίσεων του Keystone στο αρχείο *keystone.conf*. Συγκεκριμένα, προστίθενται 3 γραμμές



Στιγμιότυπο 9 - Αρχείο ρυθμίσεων *keystone.conf*

Κάθε γραμμή προκαλεί τις παρακάτω αλλαγές

- *notification_format = cadf* – Η προεπιλεγμένη μορφή ειδοποιήσεων στο DevStack είναι το "*basic*" (σε αντίθεση με το OpenStack που είναι το "*cadf*") αλλάζει σε "*cadf*"
- *notification_driver = messaging* – Οι ειδοποιήσεις δρομολογούνται προς την υπηρεσία messaging. Εκεί μπαίνουν στην ουρά του RabbitMQ που ρυθμίστηκε σε προηγούμενο βήμα
- *notification_driver = log* – Εκτός από την υπηρεσία messaging, οι ειδοποιήσεις καταγράφονται και στα αρχεία καταγραφής (logs) του *Keystone*.

Σε αυτό το σημείο πρέπει να αναφερθεί μία από ιδιαιτερότητες του DevStack. Δεν τηρούνται ξεχωριστά αρχεία καταγραφής για κάθε συστατικό του (keystone, nova, neutron κλπ). Όλες οι εγγραφές συμβάντων αποστέλλονται στο syslog όπου και αποθηκεύονται προσωρινά. Για την εφαρμογή των νέων ρυθμίσεων απαιτείται η επανεκκίνηση του web server και των υπηρεσιών του *Keystone*.

Το Keystone από τώρα και στο εξής παράγει events με τη μορφή CADF. Τα events, σύμφωνα με το documentation [43] είναι τα ακόλουθα :

Resource Type	Supported Operations	typeURI
Group	create, update, delete	data/security/group

project	create, update, delete	data/security/project
Role	create, update, delete	data/security/role
domain	create, update, delete	data/security/domain
User	create, update, delete	data/security/account/user
Trust	create, delete	data/security/trust
Region	create, update, delete	data/security/region
endpoint	create, update, delete	data/security/endpoint
service	create, update, delete	data/security/service
Policy	create, update, delete	data/security/policy
role assignment	add, remove	data/security/account/user
None	Authenticate	data/security/account/user

Πίνακας 9 - Υποστηριζόμενα events [43]

Τα παραπάνω συμβάντα θα συμπληρωθούν από τα συμβάντα που θα παράξει το Nova (compute project). Για να μπορέσει το Nova να χρησιμοποιήσει το μοντέλο CADF είναι αναγκαία η ενεργοποίηση του *audit middleware* (από το project Keystone).

Συγκεκριμένα, είναι αναγκαίες οι ακόλουθες παρεμβάσεις στο αρχείο "*api-paste.ini*" του project Nova

Στιγμιότυπο 10 - Ορισμός του audit filter και του api audit map

Στιγμιότυπο 11 - Εισαγωγή audit filter σε WSGI pipeline (μετά το authtoken) του Keystone

Το εγκατεστημένο σύστημα του DevStack έχει πλέον ενεργοποιημένες ειδοποιήσεις με τη μορφή CADF για τα υποσυστήματα *Keystone* και *Nova* και καταχωρούνται στο syslog.

8.2.3 Σενάριο ενεργειών

Το σενάριο περιλαμβάνει την εκτέλεση ενεργειών στα 2 projects που ρυθμίστηκαν να παράγουν CADF events ώστε να γίνει καταγραφή των εγγραφών στο syslog και εν συνεχεία να απομονωθούν αυτές (διότι το syslog περιλαμβάνει καταγραφές από διάφορα συστατικά του συστήματος).

Για λόγους αυτοματισμού οι ενέργειες δε θα εκτελεστούν από το web interface (Horizon) του DevStack. Στο αποθετήριο *os_helpers* δημιουργήθηκε script (*create_events.sh*) που κάνει trigger ένα σύνολο από events των 2 projects κάνοντας χρήση του python command line OpenStackClient. Πριν τη χρήση του script εκτελείται η εντολή *". openrc admin admin"* ειδικά για το DevStack. Με αυτόν τον τρόπο ορίζονται οι μεταβλητές περιβάλλοντος που σχετίζονται με το project και τα διαπιστευτήρια του χρήστη. Συγκεκριμένα για το χρήστη "admin" και το project "admin".

Keystone project	**Nova project**
• group list • role list • user list • create group • update group • add user to group • remove user from group • delete group • create role • add role to user • remove role from user • delete role • create user • disable user • delete user	• image list • flavor list • network list • security group list • create virtual machine • update virtual machine • delete virtual machine • list servers

Πίνακας 10 - Ενέργειες που εκτελεί το create_events.sh

```
#!/bin/bash

BASEDIR=$(dirname "$0")

source $BASEDIR/functions/keystone_crud
source $BASEDIR/functions/nova_crud
```

```
list_keystone_entries
crud_keystone_entries

list_nova_entries
crud_nova_entries
```

Script 4 - create_events.sh

```
#!/bin/bash

list_keystone_entries(){
      openstack group list
      openstack role list
      openstack user list
}

crud_keystone_entries(){
      #create a new group
      openstack group create --description "a testing group" --domain
default a_test_group
      #update the group
      openstack group set --description "updating the testing group" --
domain default a_test_group
      #update - add user "demo" to "a_test_group"
      openstack group add user a_test_group demo
      #update - remove user "demo" from "a_test_group"
      openstack group remove user a_test_group demo
      #delete group
      openstack group delete a_test_group

      #create a new role
      openstack role create just_a_role
      #update - add just_a_role to user demo
      openstack role add --domain default --user demo just_a_role
      #update - remove just_a_role from user demo
      openstack role  remove --domain default --user demo just_a_role
      #delete
      openstack role delete just_a_role

      #create a new user
      openstack user create --domain default --password pass --email
               ndalezios@gmail.com --description "a test user" ndalezios
      #update - disable user ndalezios
      openstack user set --disable ndalezios
      #delete
      openstack user delete ndalezios
}
```

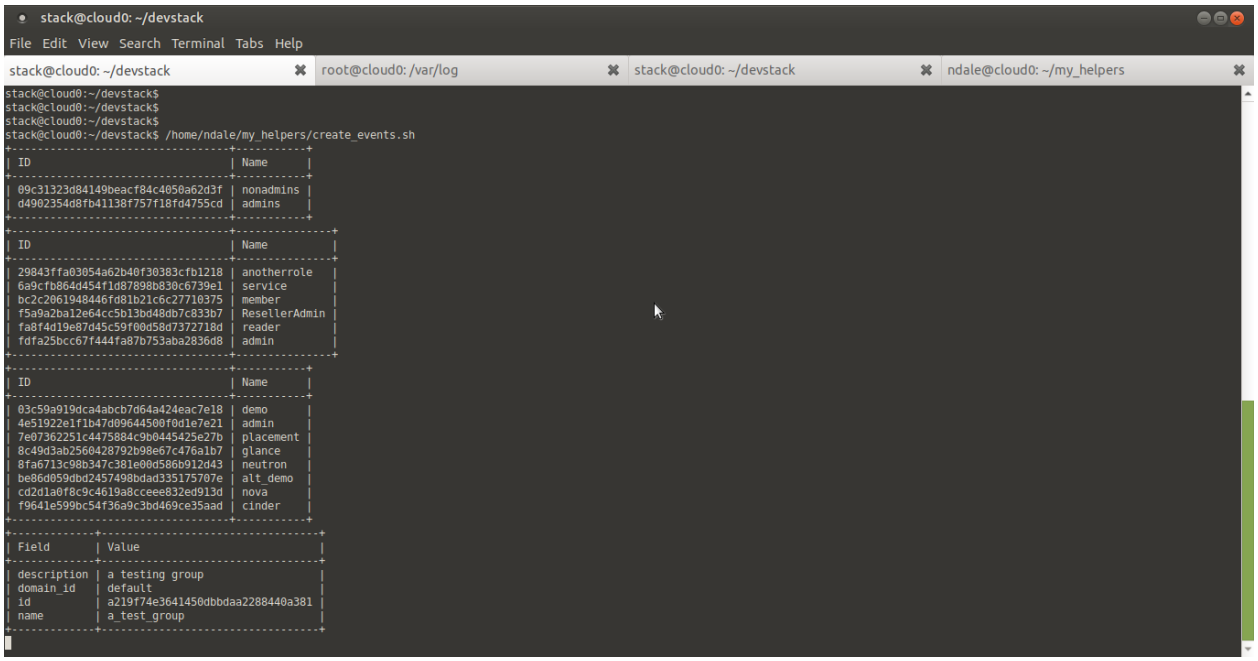
Script 5 - keystone_crud functions

```
#!/bin/bash

list_nova_entries(){
      openstack image list
      openstack flavor list
      openstack network list
      openstack security group list
}

crud_nova_entries(){
      #create a new virtual machine named "a_new_instance"
      openstack server create --flavor m1.micro --image cirros-0.3.6-
```

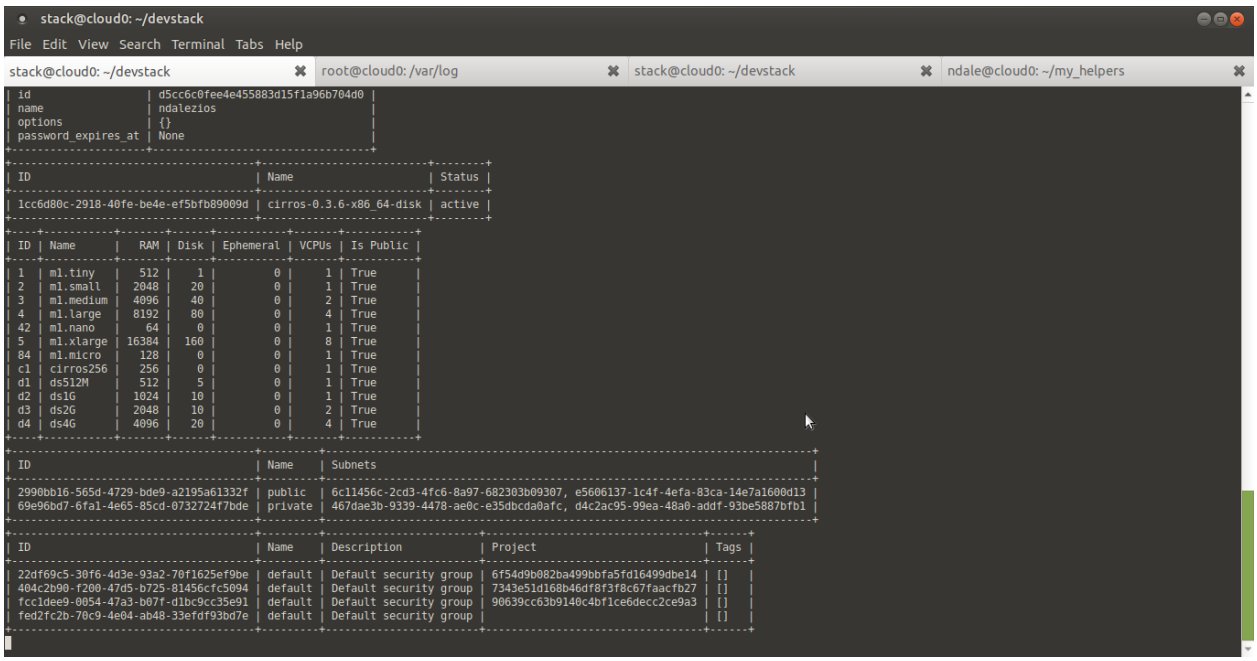
```
x86_64-disk --network private a_new_instance
      openstack server show a_new_instance
      echo "Please wait 3 minutes in order to build the virtual machine
before continuing....."
      #sleeping for 3 minutes in order to finish creating the virtual
machine
      sleep 3m
      echo "Thank you..."
      #update virtual machine "a_new_instance" by adding a description
field
      openstack server set --property Description="Just a state update"
a_new_instance
      #delete virtual machine a_new_instance
      openstack server delete a_new_instance

      #list servers to show that "a_new_instance" was deleted
      openstack server list --all-projects
}
```

Script 6 - nova_crud functions

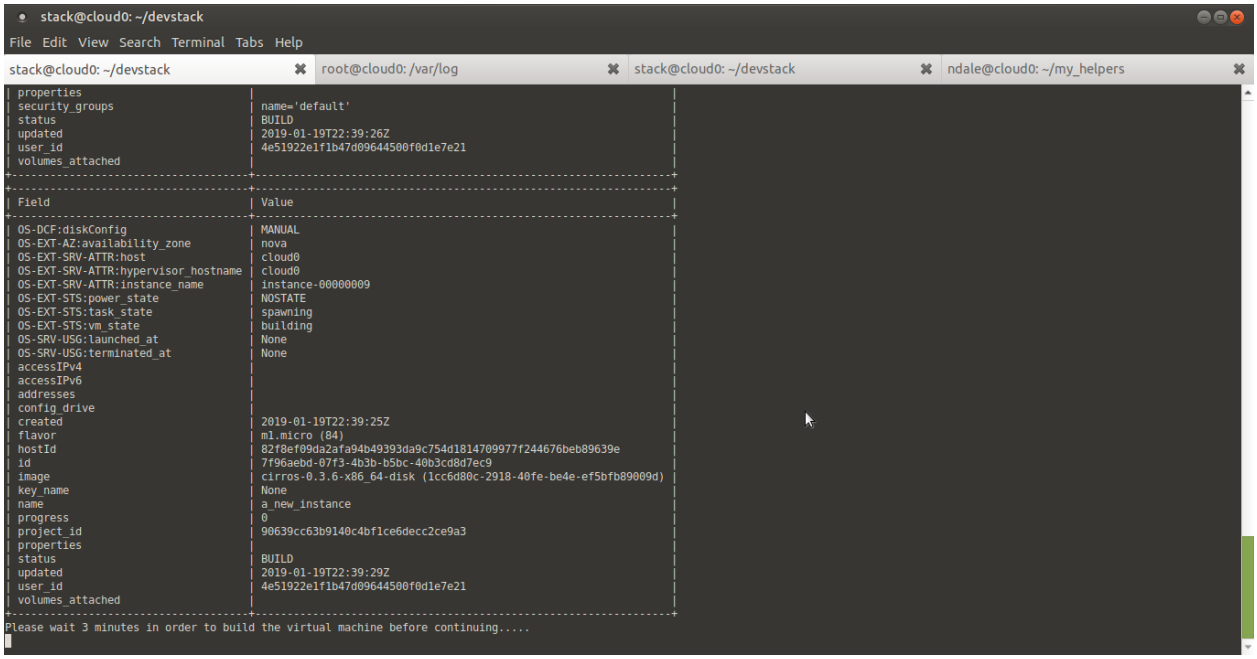
Η εκτέλεση του *create_events.sh* παράγει την ακόλουθη έξοδο



Στιγμιότυπο 12 - Έξοδος create_events.sh (1 από 3)

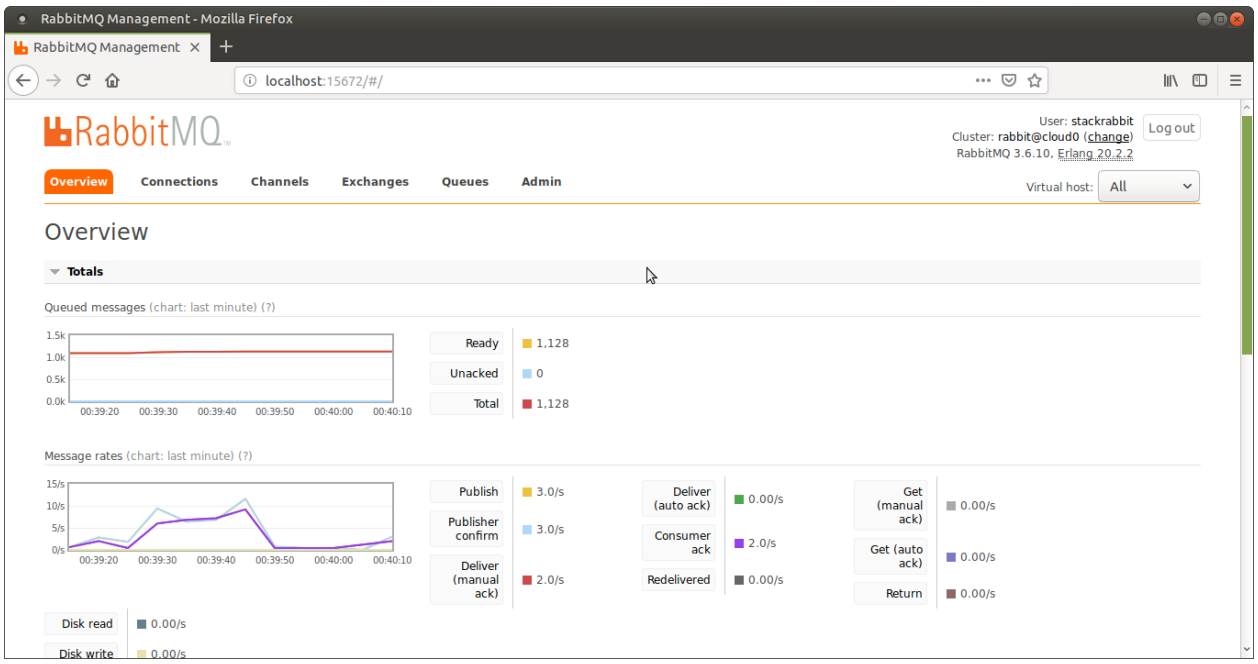


Στιγμιότυπο 13 - Έξοδος create_events.sh (2 από 3)



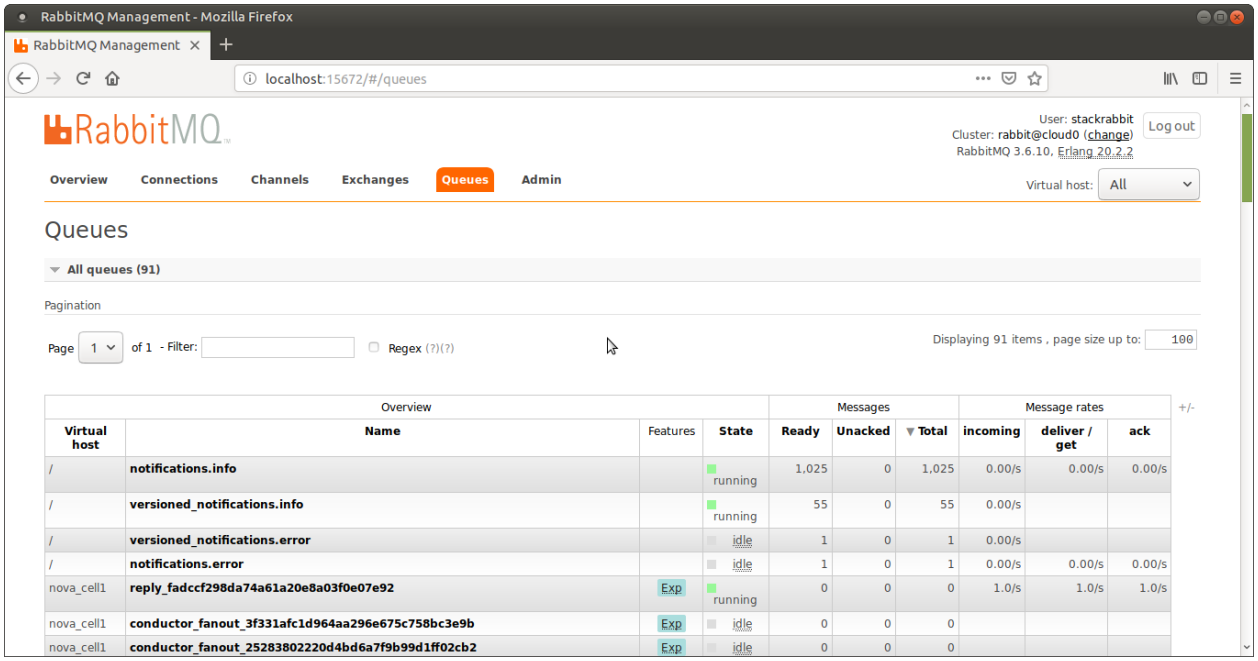
Στιγμιότυπο 14 - Έξοδος create_events.sh (3 από 3)

Κατά τη διάρκεια της εκτέλεσης του script, καταφθάνουν στο message broker event notifications μέσω μηνυμάτων. Τα μηνύματα μπορούν να γίνονται monitor από το web interface του RabbitMQ.



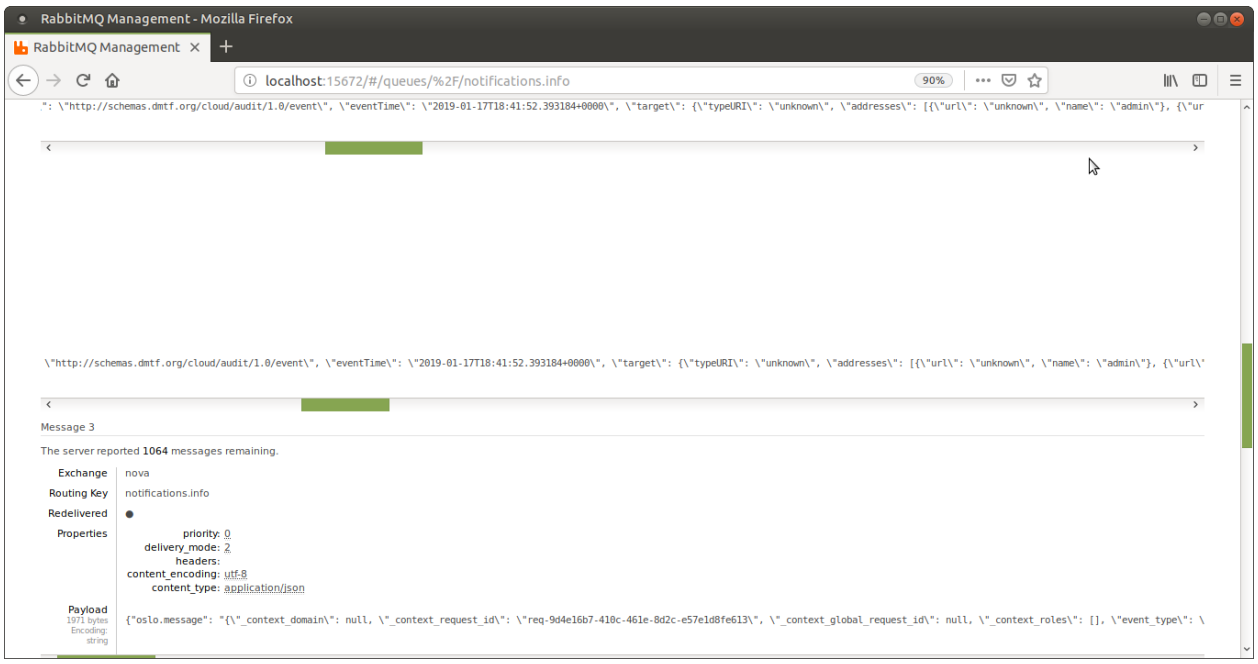
Στιγμιότυπο 15 - Σύνοψη των event notifications

Από τη διαχείριση του RabbitMQ γίνεται μετάβαση στις ουρές (queues) τις οποίες χρησιμοποιεί για αποθήκευση των μηνυμάτων.



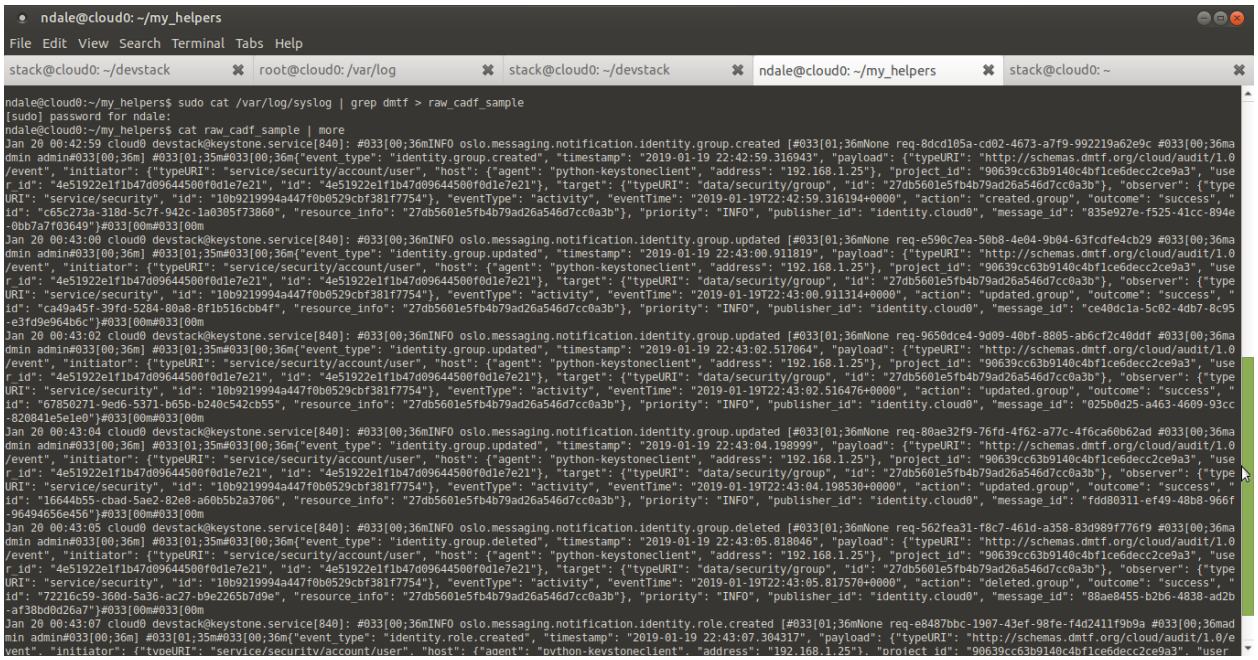
Στιγμιότυπο 16 - Ουρές μηνυμάτων

Επιλέγοντας μία ουρά και τον αριθμό των μηνυμάτων προς εμφάνιση (από το σύνολο της ουράς) μπορεί να γίνει ανάγνωση του κάθε μηνύματος



Στιγμιότυπο 17 - Εμφάνιση των 3 πιο πρόσφατων μηνυμάτων της ουράς

Τα μηνύματα όπως φαίνεται περιέχουν ειδοποιήσεις που ακολουθούν το μοντέλο CADF. Πριν την εκτέλεση του create_events.sh έγινε εκκαθάριση του syslog ώστε να μην περιέχει καταχωρήσεις από άλλα συστατικά του συστήματος. Παρόλα αυτά, γίνεται φιλτράρισμα των γραμμών που περιέχουν το ακρωνύμιο “dmtf” η παρουσία[14] του οποίου καταδεικνύει το μοντέλο CADF. Η έξοδος ανακατευθύνεται στο αρχείο “raw_cadf_sample”. Αυτό με τη σειρά του προστίθεται στο αποθετήριο *os_helpers.*



Στιγμιότυπο 18 - CADF notifications in syslog

[14] "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event"

Κεφάλαιο 9
Apache CloudStack Use Case

Το CloudStack είναι έργο ΕΛΛ/ΛΑΚ του ιδρύματος Apache. Η συμμετοχή και συνεισφορά με οποιονδήποτε τρόπο στο έργο διέπεται από κανόνες και κώδικα καλής συμπεριφοράς (http://theapacheway.com/). Σημαντική βοήθεια προσφέρουν οι λίστες ηλεκτρονικού ταχυδρομείου του CloudStack (#cloudstack, #cloudstack-dev) .Μερικοί τρόποι συνεισφοράς είναι οι μεταφράσεις (localizations), η τεκμηρίωση (documentation), ο έλεγχος για σφάλματα (testing) και οι διορθώσεις (bug fixes). Η παρούσα διατριβή έχει σκοπό την συνεισφορά στο project με την προσθήκη νέων δυνατοτήτων και συγκεκριμένα την υλοποίηση του CADF event model ως εναλλακτική του υπάρχοντος μηχανισμού καταγραφής συμβάντων.

Ο συντάκτης της διατριβής δεν ανήκει στην αποκλειστική ομάδα των Commiters του έργου αλλά σε αυτή των Non-Commiters. Οι Commiters καταχωρούν τον κώδικά τους απευθείας στο αποθετήριο του έργου ενώ οι Non-Commiters πρέπει να περάσουν τον κώδικα και τις αλλαγές τους μέσα από διαδικασία reviewing.

9.1 Προετοιμασία – Εργαλεία, Γλώσσα, Μεθοδολογία

Απαραίτητο εργαλείο είναι το git μέσω του https://github.com για τον έλεγχο των εκδόσεων. Το βασικό αποθετήριο (https://github.com/apache/cloudstack) γίνεται fork στο λογαριασμό του συντάκτη της διατριβής (https://github.com/ndalezios/cloudstack) και ακολουθεί η δημιουργία branch με το όνομα “cadf_events” (https://github.com/ndalezios/cloudstack/tree/cadf_events). Αναλυτικές οδηγίες για προγραμματιστές παρέχονται στο https://cloudstack.apache.org/developers.html. Μετά την προσθήκη και τον έλεγχο των αλλαγών στο branch “cadf_events”, γίνεται αίτημα ενσωμάτωσης στο κύριο αποθετήριο με τη δημιουργία ενός Pull Request. Με αυτόν τον τρόπο ειδοποιείται η ομάδα των Commiters για την προσθήκη. Αν η ομάδα κρίνει πως η προσθήκη ικανοποιεί συγκεκριμένες προϋποθέσεις και υλοποιεί νέες δυνατότητες τότε ενσωματώνει τις αλλαγές στο βασικό αποθετήριο του έργου.

Ο κυρίως κώδικας του CloudStack είναι γραμμένος σε Java (https://www.java.com/en/) με τη βοήθεια ενός Model View Controller (MVC) framework, του Spring (https://spring.io/). Το wiki του CloudStack (https://cwiki.apache.org/confluence/display/CLOUDSTACK) στη σελίδα

“Development 101” παρέχει αναλύτικές οδηγίες για τη δομή των πακέτων και των υποέργων του CloudStack και τις εξαρτήσεις τους, την ονοματολογία και τη δόμηση των προγραμματιστικών οντοτήτων (coding και naming conventions), το χειρισμό των εξαιρέσεων και την επικοινωνία με τη βάση δεδομένων. Παρόλο που πρόκειται για έργο ανοιχτού κώδικα είναι αναγκαία η αυστηρή τήρηση των κανόνων διότι σε διαφορετική περίπτωση δεν θα περάσει με επιτυχία το reviewing και θα μείνει απλά ένα hack σε κάποιο GitHub fork που πιθανότατα θα χαθεί στην επόμενη έκδοση.

Τέλος το εργαλείο που θα χρησιμοποιηθεί για τη μελέτη και συγγραφή του πηγαίου κώδικα είναι το IntelliJ IDEA (https://www.jetbrains.com/idea/) και συγκεκριμένα η έκδοση Community η οποία είναι διαθέσιμη δωρεάν για JVM και Android projects.

9.2 Κατανόηση - Δομή, λειτουργία, κατηγορίες Events

Το πρώτο βήμα είναι η κατανόηση του τρόπου λειτουργίας του CloudStack. Συγκεκριμένα

- Πώς γίνονται οι κλήσεις (API Calls), πώς δρομολογούνται και από ποιον και με ποιο τρόπο διεκπεραιώνονται
- Πώς καταγράφονται τα συμβάντα
- Τι θεωρείται συμβάν, ποιες πληροφορίες περιλαμβάνει και ποιες είναι οι κατηγορίες των συμβάντων
- Οι λειτουργίες της καταγραφής ενεργειών (activity logging) και της καταγραφής των συμβάντων σχετίζονται ή είναι ανεξάρτητες; Σε περίπτωση που σχετίζονται, υπάρχει επικάλυψη της πληροφορίας;

Δομικά, το CloudStack αποτελείται από έναν αριθμό έργων/υποέργων.

Project	**JAR package name**	**Πληροφορίες**
agent	cloud-agent	Αναλαμβάνει την διαχείριση σε επίπεδο πρωτοκόλλων του ServerResource (core)
Api	cloud-api	REST API, Agent API and API definitions
client	cloud-client-ui	Web interface του CloudStack
core	cloud-core	Υλοποίηση των εικονικών πόρων (hardware translation layer)
developer	cloud-developer	
engine	cloud-engine	
framework	cloudstack-framework	
plugins	cloudstack-plugins	
quickcloud	cloud-quickcloud	
server	cloud-server	Management Server (business logic)

services	cloudstack-services	
systemvm	cloud-systemvm	
test	cloud-testclient	
tools	cloud-tools	
utils	cloud-utils	Βοηθητικές μέθοδοι που μπορούν να χρησιμοποιηθούν από οποιοδήποτε project
vmware-base	cloud-vmware-base	

Πίνακας 11 - CloudStack main subprojects

Τα Events χωρίζονται σε 3 κατηγορίες

- Action events - αφορούν ενέργειες που έγιναν από το χρήστη
- Usage Events - αφορούν στατιστικά σχετικά με τη χρήση εικονικών και φυσικών πόρων
- Alerts - Ειδοποιήσεις για την κατάσταση του Management Server καθώς και για policy based events.

Επίσης χωρίζονται σε

- Standard events - Η διάρκεια της ενέργειας είναι αμελητέα και το αποτέλεσμα γνωστό αμέσως
- Long Running Job Events - Η διάρκεια της ενέργειας είναι υπολογίσιμη και το αποτέλεσμα προκύπτει σε χρόνο διαφορετικό από αυτόν της έναρξης

Τέλος διακρίνονται σε

- Synchronous Events - Προκύπτουν από ενέργειες που εκτελούνται άμεσα
- Asynchronous (Scheduled Events) - Προκύπτουν από ενέργειες που προγραμματίζονται για εκτέλεση σε συγκεκριμένη χρονική στιγμή

9.2.1 Action Events

Το CloudStack στο έργο api, στο class module *EventTypes.java* περιλαμβάνει 352 διαφορετικά Action Events. Για να γίνει απεικόνιση των Action Events με το CADF πρότυπο είναι αναγκαία η αντιστοίχιση κάθε ενός από αυτά με το ανάλογο CADF event. Όπως έχει ήδη αναφερθεί (Σχήμα 19 - Ταξινομίες της οντότητας “ACTION” [3]) το Action Taxonomy του CADF περιλαμβάνει 26 (25 ενέργειες και 1 unknown) διαφορετικές ενέργειες.

Η αντιστοίχιση αυτή αποτελεί σημαντικό πρόβλημα διότι στο CloudStack τα events περιγράφονται λεκτικά με τη μορφή συμβολοσειρών (String) συνενώνοντας ουσιαστικά τον πόρο στον οποίο συμβαίνει η ενέργεια με την ίδια την ενέργεια. Ακολουθούν μερικές ενδεικτικές περιπτώσεις

CloudStack Event	**Σημειώσεις**
VM.DESTROY	Η ενέργεια “DESTROY” δεν υπάρχει στο Action Taxonomy του CADF

NETWORK.CREATE	Δεν παρουσιάζει κάποιο πρόβλημα
PROXY.DIAGNOSTICS	Η ενέργεια “DIAGNOSTICS” δεν υπάρχει στο Action Taxonomy του CADF. Αν υπήρχε θα έπρεπε να ονομάζεται DIAGNOSE. Η σωστή αντιστοιχία είναι PROXY.DIAGNOSTICS.START
NET.RULEADD	Η ενέργεια RULEADD δεν υπάρχει στο Action Taxonomy του CADF και πρέπει να μετατραπεί σε RULE.CREATE.
CREATE_RESOURCE_DETAILS	Η ενέργεια δεν ακολουθεί συγκεκριμένη μορφή. Χρησιμοποιεί “_” και δε μπορεί να απεικονιστεί στο CADF

Πίνακας 12 - Ενδεικτικές περιπτώσεις ασυνέχειας στην ονοματοδοσία συμβάντων στο CloudStack

Από τις ανωτέρω περιπτώσεις events όπως αυτά ορίζονται στο CloudStack προκύπτει η ανάγκη χρήσης μίας ενιαίας μορφής event type αποκλειστικά για το CloudStack η οποία θα μπορούσε να γενικευτεί και σε άλλες πλατφόρμες. Η προτεινόμενη μορφή είναι η εξής

RESOURCE.(SUB-RESOURCE).(SUB-RESOURCE).ACTION

NOUN.(NOUN).(NOUN).VERB

Όπου, με κεφαλαίους λατινικούς χαρακτήρες το RESOURCE και τα SUBRESOURCES (εφόσον υπάρχουν) είναι ουσιαστικά (nouns) και το ACTION είναι ρήμα (verb). Ακολουθώντας αυτή τη μορφή, ακόμα και αν δεν υπάρχουν το Resource ή το Action στις ταξινομίες του CADF είναι εύκολο να δημιουργηθεί ένας χάρτης αντιστοίχισης συμβάντων (events map). Ειδικά για την περίπτωση που το Resource ή το Action δε μπορεί να αντιστοιχηθεί χρησιμοποιείται η τιμή “unknown”.

Στο σημείο αυτό πρέπει να επισημανθεί πως για το ειδικά για το CloudStack, όπου τα events δεν ακολουθούν κάποιο συγκεκριμένο πρότυπο, αυτά (τα events) χαρακτηρίζονται από το πεδίο *event_type* ως προς τον πόρο και την ενέργεια. Σε αντίθεση, το CADF παρέχει πεδίο eventType του οποίου οι τιμές καθορίζουν αν πρόκειται για monitor, control ή action event.

Η 1η παρέμβαση στο CloudStack αφορά στο αρχείο

cloudstack/api/src/main/java/com/cloud/event/EventTypes.java, όπου εντοπίζεται κάθε event που δεν ταιριάζει στην προτεινόμενη μορφή (noun.verb) και προστίθεται σχόλιο με το προτεινόμενο όνομα.

```
//TODO change value to VM.PASSWORD.RESET
public static final String EVENT_VM_RESETPASSWORD = "VM.RESETPASSWORD";
//TODO change value to VM.SSHKEY.RESET
public static final String EVENT_VM_RESETSSHKEY = "VM.RESETSSHKEY";
//TODO change value to ROUTER.DIAGNOSTICS.START
public static final String EVENT_ROUTER_DIAGNOSTICS = "ROUTER.DIAGNOSTICS";
```

Στιγμιότυπο 19 - Απόσπασμα EventTypes.java με προτάσεις νέας ονοματοδοσίας

Οι προτεινόμενες αλλαγές είναι συνολικά 31 και παρατίθενται αναλυτικά στο Παράρτημα Α.4 .

9.2.2 Μηχανισμός Καταγραφής Events

Η καταγραφή των συμβάντων στο CloudStack πραγματοποιείται στο package cloud-server (project server). Σε αυτό το έργο εκτελούνται οι εξής λειτουργίες :

- ανακατευθύνονται οι API κλήσεις με τη μορφή web request
- πραγματοποιούνται συγκεκριμένοι έλεγχοι στο request
- γίνεται καταγραφή σε logs των βασικών παραμέτρων του api call
- προωθείται προς την αντίστοιχη οντότητα η εντολή για την ενέργεια που περιέχεται στο request
- η οντότητα εκτελεί την ενέργεια και εναποθέτει τα αποτελέσματα σε web response
- καταγράφεται το συμβάν και αποθηκεύεται στη βάση δεδομένων

Από τα παραπάνω είναι προφανές πως το package cloud-server είναι αυτό στο οποίο πρέπει να γίνει η υλοποίηση του CADF event model καθώς περιέχει όλες τις απαραίτητες πληροφορίες που σχετίζονται με την αίτηση που προκάλεσε μία ενέργεια, την εκτέλεση της ενέργειας και το αποτέλεσμα αυτής.

Η κλάση η οποία είναι υπεύθυνη για την περιγραφή των events στο CloudStack είναι η *EventVO* (cloudstack/engine/schema/src/main/java/com/cloud/event/EventVO.java).

Τα πεδία που χρησιμοποιεί για αποθήκευση της πληροφορίας είναι :

- Id
- type
- state
- description
- createDate
- userId
- accountID
- domainId
- level
- startId
- parameters
- uuid
- archived

Από τα παραπάνω πεδία ουσιαστική πληροφορία που μπορεί να αξιοποιηθεί σε δικανική έρευνα παρέχουν τα type, state, description, createDate, userid. Το πεδίο type περιέχει μία συμβολοσειρά που αντιπροσωπεύει τον πόρο τον οποίο αφορά μία ενέργεια μαζί με την ενέργεια. Το πεδίο description είναι μία συνένωση συμβολοσειρών και πληροφοριών, διαφορετική για κάθε συμβάν, με αποτέλεσμα να μην έχει συγκεκριμένη δομή έτσι ώστε να αναλυθεί κατά λέξη (parse) από αυτοματοποιημένα εργαλεία.

Question	CloudStack EventVO fields
What	description

When	createDate
Who	uuid, userId
FromWhere	Δεν υπάρχει διαθέσιμη πληροφορία
OnWhat	description
Where	Δεν υπάρχει διαθέσιμη πληροφορία
ToWhere	Δεν υπάρχει διαθέσιμη πληροφορία

Πίνακας 13 - Ερωτήσεις που απαντά η υπάρχουσα μορφή events στο CloudStack

Από τον πίνακα γίνεται φανερό πως δεν αποθηκεύεται πληροφορία στο event σχετικά με γεωγραφική θέση, ηλεκτρονική διεύθυνση, ακριβή διεύθυνση πόρου, πλατφόρμα και πληροφορίες συστήματος που προκάλεσε το συμβάν κλπ.

9.3 Υλοποίηση CADF Event model

Κατά την υλοποίηση του CADF Event model δημιουργήθηκαν 3 top-level classes

Αρχείο	**Class name**	**Περιγραφή**
server/src/main/java /com/cloud/event/Cadf.java	Cadf	Βασικές ιδιότητες και μέθοδοι για την περιγραφή ενός CADF συμβάντος με βάση ένα CloudStack event (EventVO). Επίσης υλοποιεί βασικές λειτουργίες για αντιστοίχιση των οντοτήτων του CloudStack σε συμβατές με το CADF μοντέλο.
server/src/main/java /com/cloud/event/Resource.java	Resource	Βασικές ιδιότητες και πεδία για τον καθορισμό ενός πόρου CADF καθώς και των εξαρτώμενων από τον πόρο οντοτήτων (Credential, Addresses, Host, Geolocation, Attachments)
server/src/main/java /com/cloud/event/Taxonomies.java	Taxonomy	Απαριθμήσεις των ταξινομιών των CADF πεδίων, των CADF Resources, αντιστοίχιση με τους CloudStack πόρους και κατηγοριοποίηση των CADF ενεργειών.

Πίνακας 14 - Οι 3 νέες κλάσεις για την υλοποίηση του CADF

9.3.1 Cadf Class

Έχει μέλη τα βασικά πεδία που χρειάζονται για την καταγραφή ενός συμβάντος όπως *typeURI, eventType, action, outcome, eventTime, measurement, reason* καθώς και πεδία που είναι αντικείμενα τύπου *Resource* όπως *observer, initiator* και *target*.

Σκοπός είναι η δομή της κλάσης να είναι όμοια με αυτή του CADF model, ώστε χρησιμοποιώντας τη βιβλιοθήκη *gson* της Google (https://github.com/google/gson) να γίνει σειριοποίηση (serialization) σε κάθε αντικείμενο με τη μορφή JSON και να καταγραφεί σε αρχείο. Ουσιαστικά κάθε αντικείμενο που δημιουργείται είναι ένα event.

Στον constructor της κλάσης η παράμετρος τύπου EventVO είναι ένα αντικείμενο της υπάρχουσας μορφής event στο CloudStack.

Ορίζεται η επικεφαλίδα (*typeURI*) με την οποία το event αναγνωρίζεται ως CADF και γίνεται διαχωρισμός στο πεδίο *type* του EventVO προκειμένου να διαχωριστεί η ενέργεια από τον πόρο.

EventVO.type	Target Resource	Action
VM.CREATE	VM	CREATE
ROLE.PERMISSION.CREATE	ROLE.PERMISSION	CREATE

Πίνακας 15 - Ενδεικτικά παραδείγματα εξόρυξης του Target & Action από το EventVO

Με βάση το Action καθορίζεται και ο τύπος του CADF event (activity, monitor, control) ελέγχοντας για αντιστοίχιση σε λίστα της κλάσης Taxonomies (*eventActionToTypeMapping*). Στη συνέχεια το πεδίο-αντικείμενο Target δημιουργείται κατ' αντιστοιχία πόρων ανάμεσα στα CloudStack Resources και CADF Resources με τη βοήθεια της κλάσης Taxonomies. Επίσης αντιστοίχιση γίνεται ανάμεσα στο πεδίο *State* του EventVO και των υποστηριζόμενων τιμών του CADF (success, unknown, pending, failure). Η χρονοσήμανση του συμβάντος ορίζεται με τη μορφή *"yyyy-MM-dd'T'HH:mm:ss.Z"* στο πεδίο eventTime.

Ως Initiator ορίζεται χειροκίνητα ο CADF πόρος που αντιστοιχεί στο χρήστη ("USER") και ως Observer ο πόρος "SYSTEM.MONITOR". Στο σημείο αυτό είναι αναγκαίο να επισημανθεί πως το CADF έχει δημιουργηθεί με σκοπό να καλύπτει κάθε είδους event για κάθε πλατφόρμα. Κάποιες πλατφόρμες χρησιμοποιούν διαφορετικό πόρο για monitoring και διαφορετικό για τη διαχείριση, το CloudStack όμως δεν έχει αυτή τη δυνατότητα. Επίσης η υλοποίηση αφορά μόνο τα events που προκύπτουν από δράση του χρήστη. Για αυτό το λόγο γίνεται χειροκίνητη ανάθεση στον Initiator.

Τέλος, γίνεται χρήση κάποιων πρόσθετων πληροφοριών από το Management Server μέσω μίας λίστας τιμών οι οποίες προστίθενται στο CADF ως custom πεδία. Μερικά τέτοια πεδία είναι τα *initiator_userid* και *initiator_csAccountName* τα οποία αφορούν ειδικά το CloudStack. Άλλα πεδία είναι η IP address του Initiator, ο user-agent, η πλατφόρμα κλπ. Ο πηγαίος κώδικας της κλάσης παρατίθεται στο Παράρτημα Α.1 .

9.3.2 Resource Class

Είναι η κλάση που υλοποιεί τις ιδιότητες και τη λειτουργικότητα του πόρου (Resource). Ο πόρος χρησιμοποιείται για την αποτύπωση ενός CADF event. Αντικείμενα της κλάσης Resource είναι ο Initiator, ο Observer και ο Target. Για την περιγραφή του Resource οι

ιδιότητες – υποχρεωτικές και μη – είναι οι *typeURI, id, name, domain, credential, addresses, host, geolocation, attachments.* Ως υποκλάσεις του Resource υλοποιούνται και οι παρακάτω κλάσεις

Υποκλάση	**Περιγραφή**
Credential	Πληροφορίες για τα διαπιστευτήρια με τα οποία γίνεται η πρόσβαση στον πόρο (*type, token, authority, assertions*)
Addresses	Πληροφορίες για τη web διεύθυνση του πόρου (*url, name, port*)
Host	Πληροφορίες σχετικά με το υπολογιστικό σύστημα το οποίο αφορά ο πόρος (*id, address, agent, platform*)
Geolocation	Πληροφορίες γεοεντοπισμού του πόρου (*id, latitude, longitude, city* κλπ.)
Attachments	Περεταίρω πληροφορίες σχετικά με τον πόρο (*contentType, content, name*)
Endpoint	Πληροφορίες σχετικά με τη διαθέσιμη διεύθυνση του πόρου για χρήση μέσω web api (*url, name, port*)

Πίνακας 16 - Κλάσεις μέλη της κλάσης Resource

Ο πηγαίος κώδικας της κλάσης παρατίθεται στο Παράρτημα Α.2 .

9.3.3 Taxonomy Class

Η κλάση Taxonomy είναι βοηθητική. Περιέχει τις απαριθμήσεις τιμών (enumerations) διαφόρων πεδίων των κλάσεων CADF και Resource, όπως *EventType*, *Action*, και *Outcome*. Επίσης περιέχει τις καταχωρήσεις των λεκτικών τιμών (String constants) για τα σχετικά με την ταξινομία *Reason* πεδία και τις καταχωρήσεις λεκτικών τιμών για κάθε είδους πόρο που υποστηρίζει το CADF. Η σημαντικότερη όμως λειτουργία της κλάσης Taxonomy είναι η δημιουργία λιστών/χαρτών αντιστοίχισης ανάμεσα σε οντότητες του CloudStack και του CADF.

HashMap	**Αντιστοίχιση**
cstoCadfResourceMapping	Πόρος του CloudStack με πόρο του CADF
eventActionToTypeMapping	Ενέργεια του CloudStack με τύπο συμβάντος του CADF
eventResourcetoUuidMapping	Πόρος του CADF (όπως αντιστοιχήθηκε στο cstoCadfResourceMapping) με μοναδικό UUID

Πίνακας 17 - HashMaps της βοηθητικής κλάσης Taxonomy

Η διαδικασία της αντιστοίχισης των πόρων και των ενεργειών του CloudStack με αντίστοιχες οντότητες συμβατές με το CADF ήταν χρονοβόρα διαδικασία η οποία απαιτούσε την κατανόηση κάθε ενός από τα συμβάντα του CloudStack (352). Τα συμβάντα που δεν

κατέστη δυνατή η αντιστοίχιση τους ορίστηκαν ως "*unknown*". Ο πηγαίος κώδικας της κλάσης παρατίθεται στο Παράρτημα Α.3 .

9.3.4 Πρόσθετες παρεμβάσεις

Συνολικά οι παρεμβάσεις στον πηγαίο κώδικα του CloudStack επηρέασαν 12 αρχεία. Εκτός από τα 3 αρχεία – ομώνυμα των βασικών κλάσεων όπως αυτές αναφέρονται στον Πίνακας 14 - Οι 3 νέες κλάσεις για την υλοποίηση του CADF, οι υπόλοιπες 9 αλλαγές χωρίζονται σε 3 κατηγορίες

- Αλλαγές στα αρχεία ρυθμίσεων
- Κλήσεις και συλλογή πρόσθετων πληροφοριών
- Προτάσεις και βελτιώσεις

Αρχείο	**Αλλαγές**
build/replace.properties	Ορίζεται το property EVENTLOG=event.log, το όνομα δηλαδή του αρχείου στο οποίο θα αποθηκεύονται τα CADF events
client/conf/log4j-cloud.xml.in	Ρύθμιση του πρόσθετου Log4J,δημιουργία του απαιτούμενου Appender για επικοινωνία με το EVENTLOG και του logger με όνομα *"com.cadf.el"*
packaging/centos63/ replace.properties	Ορισμός της διαδρομής και του ονόματος αρχείου καταγραφής (event.log) για τη διανομή CentOS 6.3
packaging/centos7/ replace.properties	Ορισμός της διαδρομής και του ονόματος αρχείου καταγραφής (event.log) για τη διανομή CentOS 7
packaging/debian/ replace.properties	Ορισμός της διαδρομής και του ονόματος αρχείου καταγραφής (event.log) για τη διανομή Debian

Πίνακας 18 - Αλλαγές στα αρχεία ρυθμίσεων

Αρχείο	**Αλλαγές**
server/src/main/java /com/cloud/event/ActionEventUtils.java	Κλήση μεθόδου δημιουργίας του CADF event (*createCadfRecord*) στο ίδιο σημείο (μέθοδος *persistActionEvent*) όπου το ίδιο το CloudStack συλλέγει τις πληροφορίες και αποθηκεύει το δικό του event στη βάση δεδομένων
server/src/main/java /com/cloud/api/ApiServlet.java	Αποθήκευση πρόσθετων πληροφοριών όπως λεπτομέρειες του API Request (*host, method* etc.) αλλά και πεδίων του CloudStack όπως userid και accountName σε HashMap (Cadf.eventExtraInformation)

Πίνακας 19 - Κλήσεις και συλλογή πρόσθετων πληροφοριών

Αρχείο	**Αλλαγές**
api/src/main/java	Εισαγωγή σχολίων στον πηγαίο κώδικα με

/com/cloud/event/EventTypes.java	πρόταση για αλλαγή της ονομασίας των events που δεν ακολουθούν το προτεινόμενο από τη διατριβή πρότυπο *Resource.Action & Noun.Verb*
server/src/main/java /com/cloud/server/ManagementServerImpl.java	Εισαγωγή κώδικα για ενεργοποίηση νέου event κατά τη διαγραφή των events από τον πίνακα ελέγχου του CloudStack

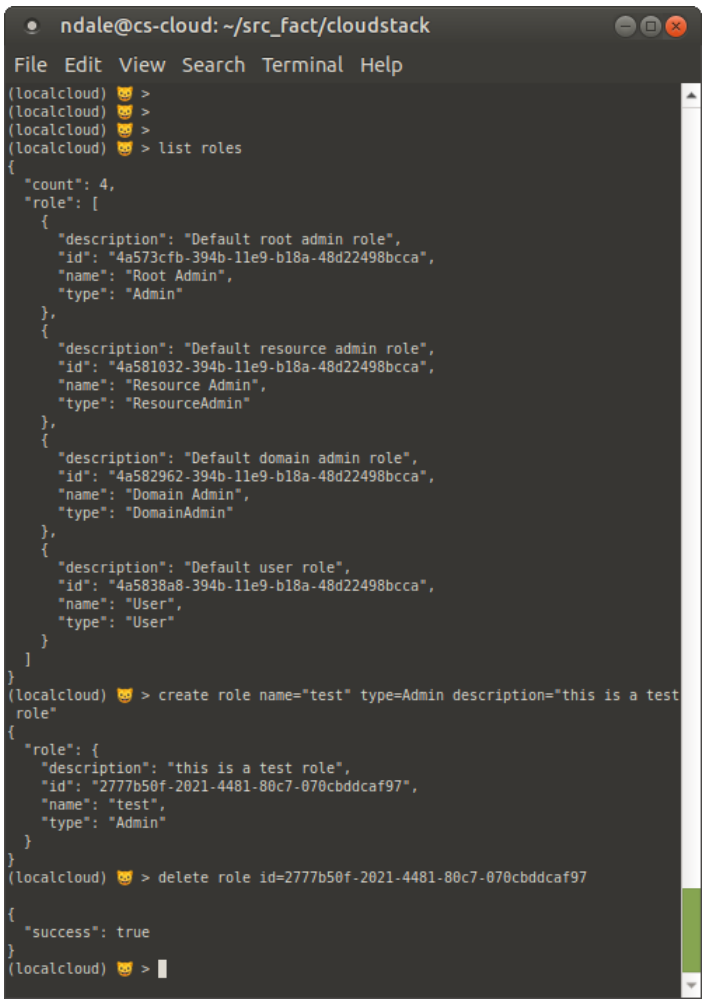
Πίνακας 20 - Προτάσεις και βελτιώσεις

9.3.5 Δοκιμή

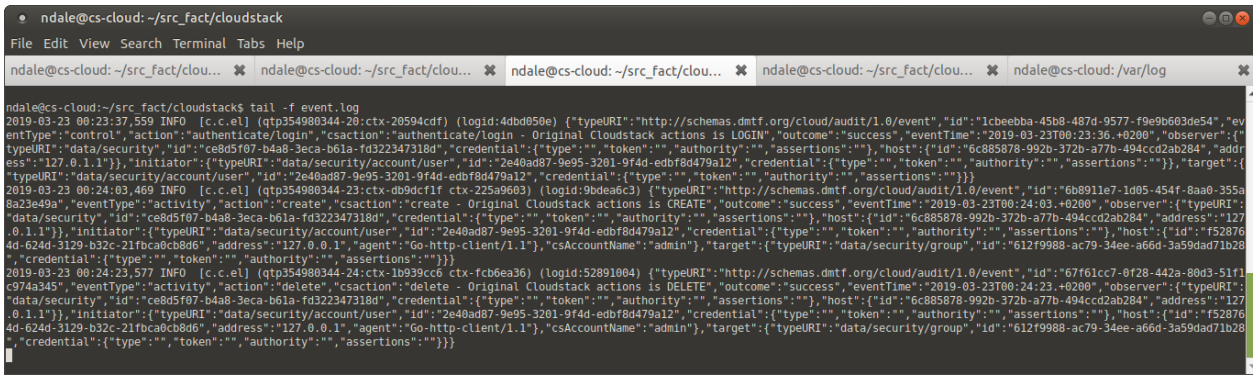
Η δοκιμή της λειτουργικότητας του μοντέλου CADF έγινε με 3 τρόπους

- Δοκιμές μέσω web interface
- Δοκιμές με χρήση του CloudMonkey (CLI) (https://github.com/apache/cloudstack-cloudmonkey)
- Με τη βοήθεια του περιβάλλοντος δοκιμών DevCloud4 (https://github.com/apache/cloudstack/tree/master/tools/devcloud4) και την εκτέλεση του python script *deployDataCenter.py* (https://github.com/apache/cloudstack/tree/master/tools/marvin/marvin)

Και στις 3 περιπτώσεις γινόταν ταυτόχρονα προβολή του αρχείου συμβάντων



Στιγμιότυπο 20 - Χρήση του CloudMonkey για δοκιμές



Στιγμιότυπο 21 - Παρακολούθηση αρχείου συμβάντων

Έχοντας ολοκληρώσει τις δοκιμές, και όντας ικανοποιημένος από τις αλλαγές πραγματοποιούνται οι εξής ενέργειες :

Οι αλλαγές γίνονται Commit στο τοπικό αποθετήριο, στο branch με όνομα "*cadf_events*" και στη συνέχεια γίνονται *push* στο απομακρυσμένο αποθετήριο στο *github* (https://github.com/ndalezios/cloudstack/tree/cadf_events)

Γίνεται αίτηση για review και ενσωμάτωση στο τρέχον αποθετήριο του CloudStack (Pull Request). Το Pull Request παίρνει τον αριθμό αναγνώρισης #3232 και βρίσκεται πλέον σε

κατάσταση αναμονής για review. Η τρέχουσα κατάσταση του pull request είναι διαθέσιμη στο σύνδεσμο https://github.com/apache/cloudstack/pull/3232.

9.4 Παρατηρήσεις και Προτάσεις

Με την ολοκλήρωση της υλοποίησης και τα 7 Ws μπορούν να απαντηθούν από το CADF σε αντίθεση με το προεπιλεγμένο μοντέλο του CloudStack (Πίνακας 13 - Ερωτήσεις που απαντά η υπάρχουσα μορφή events στο CloudStack)

Question	CADF fields
What	action, outcome
When	eventTime
Who	initiator (id, type, credential, host ip address)
FromWhere	initiator (host platform, host agent, host geolocation information)
OnWhat	target (id, type, host)
Where	Observer (id, type, host ip address)
ToWhere	Target (host platform, host agent, host geolocation information)

Πίνακας 21 - Ερωτήσεις που απαντά η υλοποίηση του CADF στο CloudStack

Η αντιπαραβολή του Πίνακας 13 - Ερωτήσεις που απαντά η υπάρχουσα μορφή events στο CloudStack και του Πίνακας 21 - Ερωτήσεις που απαντά η υλοποίηση του CADF στο CloudStack καταδεικνύει την υπεροχή του προτεινόμενου έναντι του υπάρχοντος μοντέλου.

Η λειτουργικότητα του μοντέλου CADF μπορεί να χωριστεί σε *απλή* και *προχωρημένη*. Η απλή περιορίζεται στην υλοποίηση μόνο των υποχρεωτικών πεδίων για κάθε τύπο συμβάντος ενώ η προχωρημένη περιλαμβάνει όσο περισσότερα πεδία γίνεται. Ωστόσο, τόσο στο project OpenStack (βλέπε Κεφάλαιο 8) όπως και στο Apache CloudStack το CADF δεν ήταν το μοντέλο απεικόνισης συμβάντων με το οποίο σχεδιάστηκε η πλατφόρμα. Το γεγονός αυτό οδηγεί σε κάποιες αυθαίρετες αποφάσεις σχετικά την προγραμματιστική διαδικασία.

Μία τέτοια απόφαση αφορά στο αν θα πρέπει να γίνουν αλλαγές μέσα σε κάθε ένα event ώστε να παρέχονται μέσω κώδικα οι απαραίτητες πληροφορίες για τη συμπλήρωση των πεδίων του CADF ή αν θα πρέπει – χάνοντας σε ακρίβεια – να γίνεται σε κάποιο κεντρικό σημείο αντιστοίχιση πεδίων και τιμών.

Η πιο σωστή απόφαση είναι η κατανόηση και μεταβολή κάθε ενός event ξεχωριστά. Η διαδικασία όμως έχει τα εξής μειονεκτήματα

- Είναι εξαιρετικά χρονοβόρα καθώς αφορά 352 διαφορετικά συμβάντα
- Είναι πολύπλοκη διότι κάθε event ανάλογα με το είδος του περιλαμβάνει διαφορετικές πληροφορίες

- Έχει μεγαλύτερη δυσκολία στην ενσωμάτωση καθώς για κάθε ένα event πρέπει να γίνει μεταβολή σε τουλάχιστον 1 αρχείο του CloudStack

Μόνο με αυτόν τον τρόπο δύναται να υλοποιηθεί η *προχωρημένη* μορφή του CADF event model. Λόγω του χρονικού περιορισμού της συγγραφής της διατριβής αλλά και λόγω έλλειψης ομάδας προγραμματιστών επιλέχθηκε η απλή/βασική μορφή του CADF.

Η ενότητα ολοκληρώνεται με τρεις (3) προτάσεις που υλοποιήθηκαν κατά τη διάρκεια συγγραφής της διατριβής.

a. Προσθήκη της βασικής λειτουργικότητας του μοντέλου συμβάντων CADF ως προεπιλεγμένη στην πλατφόρμα CloudStack
b. Ορισμός της διαγραφής ενός ή περισσοτέρων συμβάντων από τη βάση δεδομένων ως συμβάν. Ουσιαστικά μόνο κατά την πρώτη εγκατάσταση του CloudStack η καταγραφή των συμβάντων μπορεί να είναι κενή. Κάθε ενέργεια διαγραφής καταγραφών πρέπει να αποτελεί και η ίδια συμβάν. Μόνο με αυτόν τον τρόπο μπορεί να εντοπιστεί μία κακόβουλη διαγραφή ιστορικού ενεργειών από το CloudStack.
c. Πρόταση συγκεκριμένη ονοματολογίας για την αποτύπωση των συμβάντων, ενιαία σε ολόκληρο το project ή ακόμα καλύτερα ενιαία γενικά σε οποιοδήποτε project υλοποιεί το CADF. Η ονοματολογία πρέπει να έχει τη μορφή Resource.(Sub-resource).(Sub-resource).Action όπου το Resource είναι ουσιαστικό (noun) και το Action ρήμα (verb) και είναι όσο πιο κοντά γίνεται στις αντίστοιχες ταξινομίες και οντότητες που υποστηρίζονται από το CADF.

9.5 ACPO Principles και CADF

Όπως αναφέρεται και στο σχετικό οδηγό [44], πρέπει να τηρούνται 4 αρχές που αφορούν στις πρακτικές χειρισμού και διατήρησης των ψηφιακών πειστηρίων και έχουν κύριο αποδέκτη το προσωπικό των Αγγλικών Αστυνομικών Αρχών. Προς αποφυγή παρερμηνεύσεων παρατίθενται ως έχουν στην Αγγλική γλώσσα.

- *Principle 1: No action taken by law enforcement agencies, persons employed within those agencies or their agents should change data which may subsequently be relied upon in court*
- *Principle 2: In circumstances where a person finds it necessary to access original data, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions.*
- *Principle 3: An audit trail or other record of all processes applied to digital evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result.*
- *Principle 4: The person in charge of the investigation has overall responsibility for ensuring that the law and these principles are adhered to.*

Οι Lallie και Pimlott [10] μελετώντας την εφαρμογή των παραπάνω αρχών σε έρευνες σε public clouds επισημαίνουν μία σειρά ζητημάτων για την κάθε μία. Το CADF event model παρέχει τη δυνατότητα επίλυσης αυτών των ζητημάτων. Συγκεκριμένα,

- Σχετικά με την 1η αρχή, εντοπίζουν την ύπαρξη ενός χρονικού κενού ανάμεσα στο συμβάν και στην έναρξη της έρευνας, όπου ουσιαστικά δεν υπάρχει κανένας έλεγχος πάνω στα δεδομένα. Η λύση του auditing όμως και του CADF event logging μπορεί να επιλύσει το ζήτημα αυτό δίνοντας τη δυνατότητα στον ερευνητή να συλλέξει αρχεία καταγραφής από ένα χρονικό σημείο Α (πριν το ερευνώμενο συμβάν) μέχρι ένα χρονικό σημείο Β (τη στιγμή της έναρξης της έρευνας). Με αυτόν τον τρόπο αν και δεν υπάρχει προστασία των δεδομένων από αλλοιώσεις ωστόσο υπάρχει καταγραφή αυτών των αλλοιώσεων (αν υπάρχουν).
- Σχετικά με τη 2η αρχή, εφόσον πρόκειται μόνο για αρχεία καταγραφής επιτρέπεται η πρόσβαση μόνο για ανάγνωση. Εξάλλου το “Acquisition” αποτελεί τυπική και καθορισμένη διαδικασία σε κοινά (όχι cloud) συστήματα. Άρα δεν αυξάνεται το απαιτούμενο επίπεδο εξειδίκευσης (competency). Μάλιστα η απόκτηση των αρχείων καταγραφής δύναται να γίνει από τον ίδιο το CSP. Σε περίπτωση όμως που συμβεί οποιαδήποτε αλλαγή στα αρχεία καταγραφής, η αλλαγή αυτή πρέπει να θεωρηθεί ως ένα συμβάν και να καταγραφεί. Ειδικότερα, στην ερευνώμενη πλατφόρμα (CloudStack) η διαδικασία της διαγραφής των καταγεγραμμένων events ΔΕΝ αποτελεί συμβάν και δεν καταγράφεται. *Ο συντάκτης της διατριβής προτείνει την δημιουργία νέου συμβάντος και καταγραφής για τη συγκεκριμένη ενέργεια*.
- Ως προς την 3η αρχή περί παροχής audit trail για κάθε ενέργεια πάνω στα ψηφιακά πειστήρια έτσι ώστε μία τρίτη ερευνητική οντότητα να έχει τα ίδια αποτελέσματα εκτελώντας τις ίδιες ενέργειες η θέση της διατριβής είναι η εξής : Λόγω της απουσίας ενοποιημένης μορφής αρχείων καταγραφής ο ερευνητής καταφεύγει πολλές φορές σε custom λύσεις ανάλογα με την ερευνώμενη πλατφόρμα (OpenStack, CloudStack, AWS, MS Azure). Η διαδικασία επανάληψης των ενεργειών προκειμένου να επιτευχθεί το ίδιο αποτέλεσμα περιλαμβάνει την ανάπτυξη αυτής της custom λύσης. Η απάντηση έρχεται με την καθιέρωση συγκεκριμένου προτύπου καταγραφής και τη δημιουργία τυποποιημένων εργαλείων - είτε ανοιχτού κώδικα (όπου είναι ευκολότερο να ελεγχθεί αν επηρεάζουν την ακεραιότητα των πειστηρίων) είτε εμπορικών λύσεων. Ένα εργαλείο για όλες τις πλατφόρμες μπορεί περνώντας τους απαραίτητους ελέγχους να πιστοποιηθεί από το NIST.

9.6 CADF Consumers

Η καθιέρωση ενός προτύπου καταγραφής συμβάντων απαλλάσσει τους προγραμματιστές από την ανάγκη μελέτης και παραμετροποίησης των εργαλείων ανάλυσης συμβάντων ανάλογα με την πλατφόρμα που παρήγαγε τις καταγραφές. Έτσι επικεντρώνονται αποκλειστικά στο πρότυπο και την ερμηνεία των πεδίων του. Μία τέτοια προσπάθεια αποτελεί το εργαλείο C.Lo.D (CADF Log Detective), γραμμένο σε Python3, που είναι

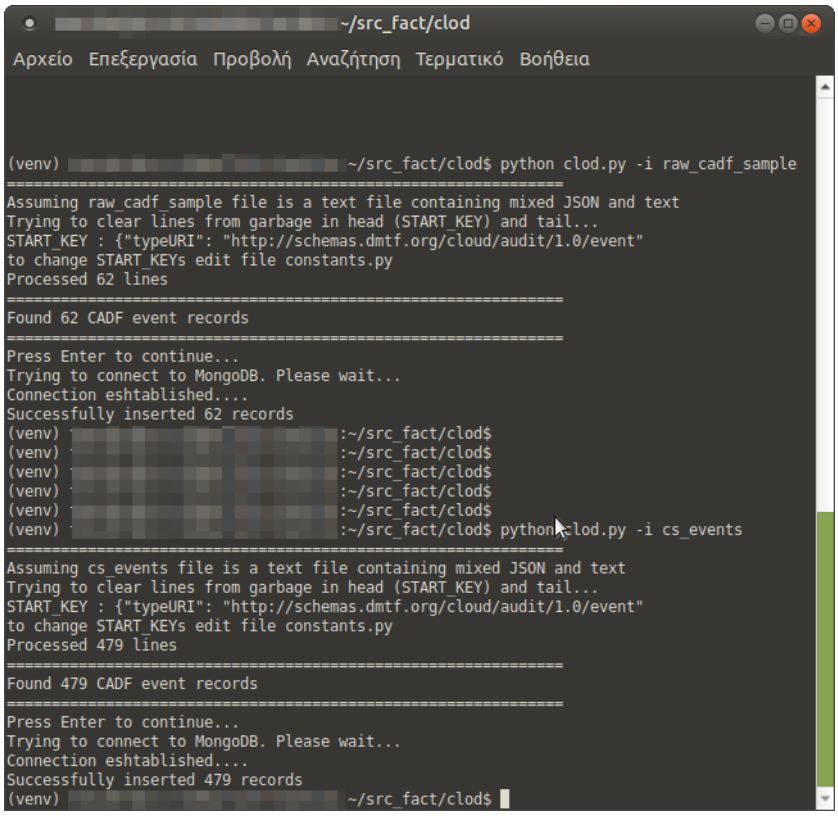
διαθέσιμο υπό την άδεια Apache 2.0 σε δημόσιο αποθετήριο στο GitHub (https://github.com/ndalezios/clod).

Το Clod δέχεται στην είσοδο ένα αρχείο καταγραφής το οποίο περιέχει καταγραφές με τη μορφή CADF σε JSON. Κάθε γραμμή στο αρχείο μπορεί πριν και μετά την καταγραφή να περιλαμβάνει πληροφορίες της πλατφόρμας (platform specific information).

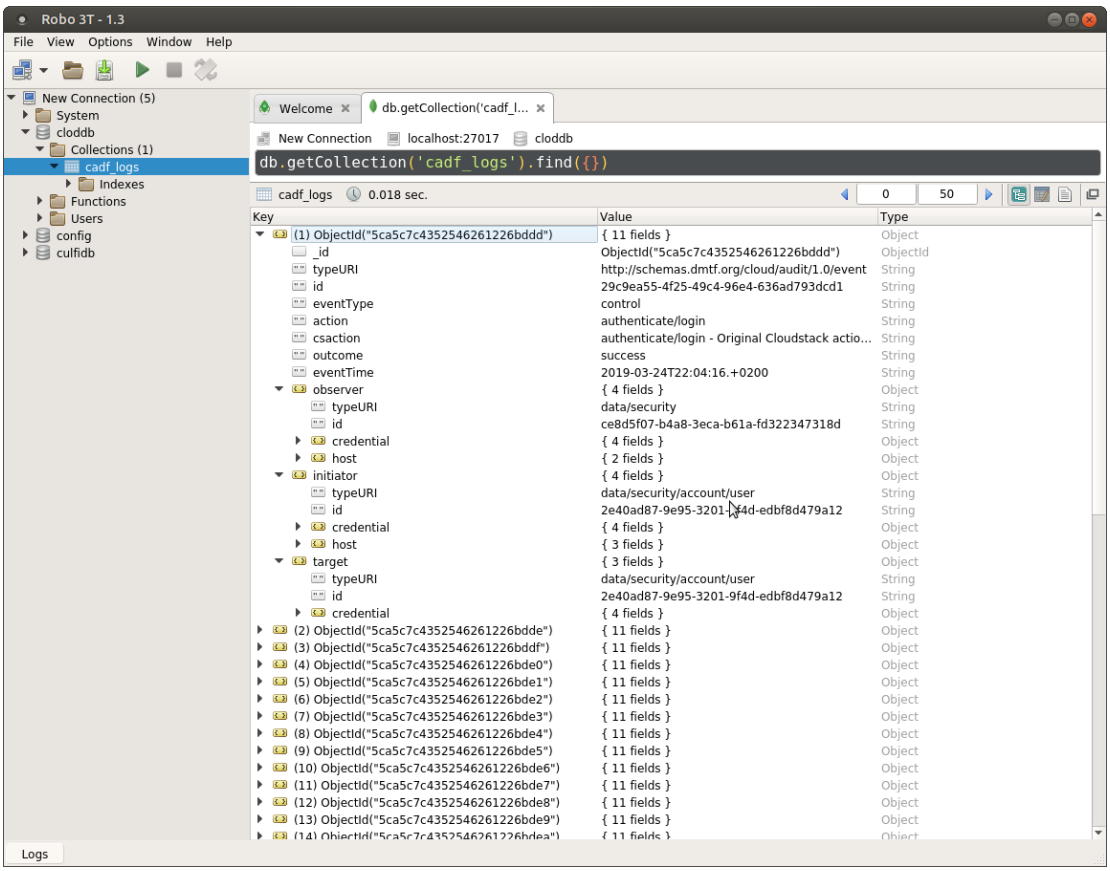
Leading data (platform specific)	**CADF data**	**Trailing data (platform specific)**

Σχήμα 26 - Δομή καταχώρησης αρχείου καταγραφής

Το εργαλείο για κάθε γραμμή αναζητά το αναγνωριστικό του CADF event (typeURI – βλέπε Πίνακας 5 - Τα πεδία του CADF Event Record). Μόλις το εντοπίσει το απομονώνει από το σημείο που εντοπίστηκε μέχρι και το σημείο όπου ο αριθμός των συμβόλων "{" και "}" είναι ίσος. Με αυτόν τον τρόπο μόνο τα δεδομένα του CADF αποθηκεύονται σε μία python list. Αυτή με τη σειρά της μπορεί εύκολα να μετατραπεί σε JSON array. Τα CADF συμβάντα στη συνέχεια αποθηκεύονται σε NoSQL database. Συγκεκριμένα επιλέγεται διαχειριστής βάσης τύπου "document store" όπου χειρίζεται ιδανικά JSON documents. Ο διαχειριστής είναι η ανοιχτού κώδικα έκδοση του MongoDB Community Server (https://www.mongodb.com/download-center/community). Από το σημείο αυτό κι έπειτα, ο ερευνητής μπορεί με ένα MongoDB client ή χρησιμοποιώντας κώδικα να συντάξει και να εκτελέσει ερωτήματα πάνω στα αποθηκευμένα CADF συμβάντα.



Στιγμιότυπο 22 - Εκτέλεση CLoD



Στιγμιότυπο 23 - Συμβάντα αποθηκευμένα σε NoSQL από το CLoD

Κεφάλαιο 10
Επίλογος

Η παρούσα μεταπτυχιακή διατριβή αφού πραγματοποίησε μία σύντομη επισκόπηση στην τεχνολογία του cloud, εμβάθυνε στον τομέα της δικανικής ανάλυσης στο cloud, τομέας που όπως αποδεικνύεται και από την μελέτη της βιβλιογραφίας είναι ακόμα σε πρώιμο στάδιο. Συγκεκριμένα καταπιάστηκε με το ζήτημα της ενοποίησης των αρχείων καταγραφής στο cloud αναλύοντας τις υπάρχουσες λύσεις σε 2 πλατφόρμες ανοιχτού κώδικα, το OpenStack και το Apache CloudStack. Στην πλατφόρμα OpenStack πλέον η προεπιλεγμένη μορφή απεικόνισης συμβάντων είναι το μοντέλο CADF. Το μοντέλο CADF, αναλύθηκε εις βάθος παραθέτοντας ταυτόχρονα τους λόγους για τους οποίους είναι προτιμώμενο έναντι των προκατόχων του

10.1 Συμπεράσματα

Το μοντέλο απεικόνισης συμβάντων CADF είναι ένα απλό στη δομή αλλά ταυτόχρονα πλούσιο ως προς την αναπαριστώμενη πληροφορία και ταυτόχρονα επεκτάσιμο. Υπερτερεί έναντι άλλων μοντέλων δίνοντας απαντήσεις και στα 7 W's της δικανικής έρευνας. Επίσης, παρέχει ένα τέτοιο isolation level ώστε δεν αποκαλύπτει καμία πληροφορία για την υποδομή στην οποία λαμβάνουν χώρα τα συμβάντα, τόσο σε επίπεδο υλικού όσο και σε επίπεδο λογισμικού. Στην πλατφόρμα OpenStack, παρόλο που πλέον είναι η προεπιλεγμένη μορφή απεικόνισης συμβάντων, το CADF δε χρησιμοποιεί ολόκληρο το εύρος των δυνατοτήτων του, γεγονός που δεν ήταν κατανοητό μέχρι την ολοκλήρωση της παρούσας διατριβής. Το μοντέλο CADF αναπτύχθηκε στην πιο απλή του μορφή – δηλαδή χρησιμοποιώντας μόνο τα υποχρεωτικά πεδία – για την πλατφόρμα Apache CloudStack. Παρόλα αυτά, παρέχει αρκετά πλουσιότερη σε δικανική αξία πληροφορία σε σχέση με το αρχικό μοντέλο του CloudStack χωρίς να προκαλεί παρενέργειες στη λειτουργία ολόκληρου του project. Ο λόγος για την μη πλήρη υλοποίηση είναι προφανώς και ο ίδιος λόγος που δεν έγινε αυτή και το OpenStack. Η άντληση όλων των πληροφοριών για την τροφοδότηση των πεδίων του CADF απαιτεί παρέμβαση στον κώδικα του κάθε ενός συμβάντος της πλατφόρμας ξεχωριστά – ήτοι σχεδόν σε κάθε αρχείο του έργου. Στο CloudStack τα συμβάντα είναι 352, στο OpenStack είναι ακόμη περισσότερα. Οι αλλαγές και οι παρενέργειες που θα επιφέρει μία τέτοια παρέμβαση απαιτούν εξονυχιστικό έλεγχο με χρονοβόρες διαδικασίες. Η ολοκληρωμένη λειτουργικότητα του CADF σε οποιοδήποτε project δε μπορεί να προστεθεί ως ένα απλό enhancement. Είναι αναγκαίος ο επανασχεδιασμός μεγάλου τμήματος του έργου που αφορά.

Η χρήση ενός προτύπου όπως το CADF ευνοεί τη δημιουργία εργαλείων ανάλυσης των πληροφοριών που αναπαριστούν τα πρότυπα, απαλλάσσοντας τους προγραμματιστές από το άγχος της πλατφόρμας. Το εργαλείο C.Lo.D. αποτελεί τέτοιο παράδειγμα καθώς δέχεται ως είσοδο αρχεία καταγραφής σε πρότυπο CADF τόσο από την πλατφόρμα OpenStack όσο και από την πλατφόρμα CloudStack.

Κάθε έργο το οποίο μπορεί να γίνει αντικείμενο επίθεσης ή εκμετάλλευσης από κακόβουλους χρήστες είναι αναγκαίο κατά το σχεδιασμό του να προβλέψει να είναι όσο περισσότερο forensically friendly γίνεται. Η ομάδα των προγραμματιστών είναι αναγκαίο να συνεργάζεται με την ομάδα security όχι μόνο σε θέματα ελέγχου για ευπάθειες αλλά κυρίως για το τι πληροφορία θα συλλεχθεί σε περίπτωση επιθυμητής και μη επιθυμητής χρήσης ενός project.

10.2 Μελλοντική Δουλειά

Για τον περεταίρω έλεγχο της υλοποίησης του CADF στην πλατφόρμα Apache CloudStack θα πρέπει να εκτελεστεί μία σειρά σεναρίων με αντίστοιχη δικανική ανάλυση. Ένα ενδεικτικό παράδειγμα είναι η προσομοίωση ενός κακόβουλου λογισμικού τύπου ransomware το οποίο μέσω ενός ανώνυμου λογαριασμού να δημιουργεί στο CloudStack εικονικές μηχανές με σκοπό να τις χρησιμοποιήσει ως Command and Control Centers. Η δικανική έρευνα θα πρέπει να αναλύσει τις πληροφορίες από τα CADF αρχεία καταγραφής. Με αυτόν τον τρόπο θα εντοπιστούν τυχόν παραλήψεις αλλά κυρίως θα έρθουν στην επιφάνεια πεδία που θα έπρεπε να περιέχουν πληροφορίες έτσι ώστε να βοηθηθεί η έρευνα. Το ίδιο use case scenario θα πρέπει να εξεταστεί και στην πλατφόρμα OpenStack. Όλοι οι εμπλεκόμενοι με μία δικανική έρευνα φορείς, όπως Αστυνομία, ερευνητές, Δίωξη Ηλεκτρονικού Εγκλήματος, Δικαστικές Αρχές κλπ. θα πρέπει να εξετάσουν αν μπορούν να χρησιμοποιήσουν και να κάνουν αποδεκτά τα όποια στοιχεία προκύπτουν από την ανάλυση του CADF καθώς και αν αυτά αρκούν. Ακόμα, η υλοποίηση του CADF από OpenStack αλλά και CloudStack μπορεί να χρησιμοποιηθεί για τη δημιουργία datasets με πληροφορίες συμβάντων, κανονικής δραστηριότητας αλλά και ύποπτης δραστηριότητας με σκοπό να αποτελέσουν είσοδο σε αλγορίθμους Machine Learning. Τέλος, το μοντέλο απεικόνισης CADF θα μπορούσε να μεταφερθεί/υλοποιηθεί και στην τρίτη σε χρήση πλατφόρμα ανοιχτού κώδικα IaaS cloud, το OpenNebula.

Παράρτημα Α
Πηγαίος Κώδικας

A.1 Cadf.class

```
// Licensed to the Apache Software Foundation (ASF) under one
// or more contributor license agreements.  See the NOTICE file
// distributed with this work for additional information
// regarding copyright ownership.  The ASF licenses this file
// to you under the Apache License, Version 2.0 (the
// "License"); you may not use this file except in compliance
// with the License.  You may obtain a copy of the License at
//
//   http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied.  See the License for the
// specific language governing permissions and limitations
// under the License.

package com.cloud.event;

import com.cloud.exception.InvalidParameterValueException;
import com.google.gson.annotations.Expose;
import org.apache.log4j.Logger;

import java.net.InetAddress;
import java.net.UnknownHostException;
import java.text.SimpleDateFormat;
import java.util.HashMap;
import java.util.UUID;

public class Cadf {

    //private class variables have no underscore prefix (as instructed in
Coding conventions document -
    // Naming Conventions - Instruction 6) because the whole class is being
logged and is more readable
    // without underscores
    @Expose (serialize = false)
    private static final Logger s_logger = Logger.getLogger(Cadf.class);

    @Expose
    private String typeURI;
    @Expose
    private String id;
    @Expose
    private String eventType;
    @Expose
    private String action;
    @Expose (serialize = false)
    private Taxonomies.Action _tmpAction;

    @Expose
```

```
    private String csaction; //CloudStack original Action
    @Expose
    private String outcome;
    @Expose
    private String eventTime;

    @Expose
    private Resource observer;
    @Expose
    private Resource initiator;
    @Expose
    private Resource target;

    //TODO
    private String measurement;
    private String reason;

    @Expose (serialize = false)
    public static HashMap<String, String> eventExtraInformation = new
HashMap<String, String>();

    /**
     * @param event is the generic CS event
     */
    public Cadf(EventVO event) {
        String eventtarget;
        String eventaction;

        typeURI = "http://schemas.dmtf.org/cloud/audit/1.0/event";

        //Event unique identifier
        id = event.getUuid();

        if (!event.getType().contains(".")) {
            eventaction = Taxonomies.Action.UNKNOWN.getValue();
            eventtarget = Taxonomies.Resource.UNKNOWN;
        } else {
            //Substrings "VM" from VM.CREATE, "ROLE.PERMISSION" from
ROLE.PERMISSION.CREATE etc
            eventtarget = new String(event.getType().substring(0,
event.getType().lastIndexOf(".")));

            //Substrings "CREATE" from VM.CREATE, "CREATE" from
ROLE.PERMISSION.CREATE etc.
            // +1 is used to ignore the "." before substring
            eventaction = new
String(event.getType().substring(event.getType().lastIndexOf(".") + 1));

            //eventtarget is event's Target
            //eventaction is event's Action
        }

        //sets EventType, Action and Target
        //Answers to WHAT, ONWHAT, TOWHERE
        setCADFAction(eventaction);

        eventType = getCADFEventType();


        //Answers to ONWHAT and TOWHERE
        target = new Resource(getCADFResourceName(eventtarget),
getCADFResourceUUID(eventtarget));
```

```
        //sets Outcome
        //Answers to WHAT
        mapEventStateToCADFTaxonomy(event.getState().toString());

        //must add reason

        //Answers to WHEN
        //_eventTime must be in UTC format
        if (event.getCreateDate() != null) {
            eventTime = new SimpleDateFormat("yyyy-MM-
dd'T'HH:mm:ss.Z").format(event.getCreateDate());
        } else {
            eventTime="";
        }

        //Answers to WHO and FROMWHERE
        //"USER" CS Resource corresponds to
Taxonomies.Resource.DATA_SECURITY_ACCOUNT_USER CADF Resource

        initiator = new
Resource(Taxonomies.Resource.DATA_SECURITY_ACCOUNT_USER,
                getCADFResourceUUID("USER"));


        //Answers to WHERE

        observer = new Resource(Taxonomies.Resource.DATA_SECURITY,
                getCADFResourceUUID("SYSTEM.MONITOR"));

        try {
            InetAddress inetAddress = InetAddress.getLocalHost();
            observer.host = new Resource.Host();
            observer.host = new
Resource.Host(UUID.nameUUIDFromBytes(inetAddress.getHostName().getBytes()).
toString(),
                    inetAddress.getHostAddress(), null , null);

        } catch (UnknownHostException e) {
            s_logger.error(e.getMessage());
        }
        if (eventExtraInformation != null &&
                eventExtraInformation.get("initiator_host") != null &&
                eventExtraInformation.get("initiator_user-agent") != null
&&
                !eventExtraInformation.get("initiator_host").isEmpty()) {
            initiator.host = new
Resource.Host(UUID.nameUUIDFromBytes(eventExtraInformation.get("initiator_h
ost").getBytes()).toString(),
                    eventExtraInformation.get("initiator_host"),
                    eventExtraInformation.get("initiator_user-agent"),
                    null);
        }
        if (eventExtraInformation != null &&
                eventExtraInformation.get("initiator_userid") != null &&
                !eventExtraInformation.get("initiator_userid").isEmpty()) {
            initiator.userid =
eventExtraInformation.get("initiator_userid");
        }
        if (eventExtraInformation != null &&
                eventExtraInformation.get("initiator_csAccountName") !=
null &&
```

```
!eventExtraInformation.get("initiator_csAccountName").isEmpty()) {
            initiator.csAccountName =
eventExtraInformation.get("initiator_csAccountName");
        }
    }


    /**
     * Maps event state to CADF Outcome Taxonomy and sets outcome
     * to a CADF compatible value
     *
     * @param eventstate is the String value of CS Event State
     *                   use event.getState().asString
     */
    private void mapEventStateToCADFTaxonomy(String eventstate) {
        switch (eventstate) {
            case "Completed" :
                outcome = Taxonomies.Outcome.SUCCESS.getValue();
                break;
            case "Created" :
            case "Scheduled" :
                outcome = Taxonomies.Outcome.UNKNOWN.getValue();
                break;
            case "Started" :
                outcome = Taxonomies.Outcome.PENDING.getValue();
                break;
            default :
                outcome = Taxonomies.Outcome.FAILURE.getValue();
                break;
        }
    }


    /**
     * Maps eventaction to a CADF Action and sets action to a CADF
compatible value
     *
     * @param eventaction CloudStack's action for every event type. eg for
eventType
     *                    ROLE.PERMISSION.CREATE action is CREATE.
     */
    private void setCADFAction(String eventaction) {
        Boolean isFound = false;
        for (Taxonomies.Action ta : Taxonomies.Action.values() ) {
            if (ta.getValue().equalsIgnoreCase(eventaction)) { //exact
match
                _tmpAction = ta;
                action = ta.getValue();
                csaction = ta.getValue() + " - Original Cloudstack actions
is " + eventaction;
                isFound = true;
                break;
            } else if (ta.getValue().contains(eventaction.toLowerCase())) {
//partial match
                _tmpAction = ta;
                action = ta.getValue();
                csaction = ta.getValue() + " - Original Cloudstack actions
is " + eventaction;
                isFound = true;
                break;
            }
        }
        if (!isFound) {
```

```
            _tmpAction = Taxonomies.Action.UNKNOWN;
            action = Taxonomies.Action.UNKNOWN.getValue();
            csaction = Taxonomies.Action.UNKNOWN.getValue() + " - Original
Cloudstack action is " + eventaction;
        }
    }

    /**
     * Maps eventType to a CADF compatible value based on the events Action
     */
    private String getCADFEventType() {
        Taxonomies.EventType _tmpEventType;
        _tmpEventType =
Taxonomies.eventActionToTypeMapping.get(_tmpAction);
        if (_tmpEventType == null) {
            _tmpEventType = Taxonomies.EventType.MONITOR;
        }
        return _tmpEventType.getValue();
    }

    /**
     * Maps CloudStack Resource to a CADF compatible value and returns the
name of the resource
     *
     * @param csResource is a String representation of CS Resource that
started the event
     *                   eg for eventType ROLE.PERMISSION.CREATE csResource
is ROLE.PERMISSION
     */
    private String getCADFResourceName(String csResource) {
        String tmpResourceName =
Taxonomies.cstoCadfResourceMapping.get(csResource);
        if (tmpResourceName == null || tmpResourceName.isEmpty()) {
            tmpResourceName = Taxonomies.Resource.UNKNOWN;
        }
        return tmpResourceName;
    }

    /**
     * Maps CloudStack Resource to a CADF compatible value and returns the
UUID value
     *
     * @param csResource is a String representation of CS Resource that
started the event
     */
    private String getCADFResourceUUID(String csResource) {
        String tmpResourceUUID =
Taxonomies.eventResourcetoUuidMapping.get(csResource);
        if (tmpResourceUUID == null || tmpResourceUUID.isEmpty()) {
            tmpResourceUUID =
UUID.nameUUIDFromBytes("UNKNOWN".getBytes()).toString();
        }
        return tmpResourceUUID;
    }

    /**
     * Checks the number and name of mandatory fields differs depending on
the EventType .
     * @throws InvalidParameterValueException
     */
    public void checkMandatoryFields() {
        if (eventType.equals(Taxonomies.EventType.MONITOR.getValue())) {
            if ((initiator == null) || (action == null) || (target == null)
```

```
|| (outcome == null) ||
                    (measurement == null)) {
                throw new InvalidParameterValueException("Initiator,
Action, Target, Outcome, Measurement " +
                        "fields are mandatory for MONITOR events");
            }
        } else if
(eventType.equals(Taxonomies.EventType.CONTROL.getValue())) {
            if ((initiator == null) || (action == null) || (target == null)
|| (outcome == null) ||
                    (reason == null) || (measurement == null)) {
                throw new InvalidParameterValueException("Initiator,
Action, Target, Outcome, Reason, Measurement " +
                        "fields are mandatory for CONTROL events");
            }
        } else {
            //if
(eventType.equals(Taxonomies.EventType.ACTIVITY.getValue())) {
                if ((initiator == null) || (action == null) || (target ==
null) || (outcome == null) ||
                        (measurement == null)) {
                    throw new InvalidParameterValueException("Initiator,
Action, Target, Outcome, Measurement " +
                            "fields are mandatory for ACTIVITY events");
                }
        }
    }
}
```

A.2 Resource.java

```
// Licensed to the Apache Software Foundation (ASF) under one
// or more contributor license agreements.  See the NOTICE file
// distributed with this work for additional information
// regarding copyright ownership.  The ASF licenses this file
// to you under the Apache License, Version 2.0 (the
// "License"); you may not use this file except in compliance
// with the License.  You may obtain a copy of the License at
//
//   http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied.  See the License for the
// specific language governing permissions and limitations
// under the License.

package com.cloud.event;

import com.cloud.exception.InvalidParameterValueException;
import com.google.gson.annotations.Expose;

public class Resource {
    @Expose
    private String typeURI; //MANDATORY
    @Expose
    private String id; //MANDATORY
    @Expose
    private String name; //OPTIONAL
    @Expose
    private String domain; //OPTIONAL
    @Expose
    private Credential credential; //OPTIONAL //subtype
    @Expose
    private String addresses; //OPTIONAL //subtype
    @Expose
    protected Host host; //OPTIONAL //subtype
    @Expose
    private String geolocation; //OPTIONAL //subtype
    @Expose
    private String geolocationId; //OPTIONAL //subtype
    @Expose
    private String attachements; //OPTIONAL //subtype
    @Expose
    protected String userid;
    @Expose
    protected String csAccountName;

    public static class Credential {
        @Expose
        private String type;
        @Expose
        private String token;
        @Expose
        private String authority;
        @Expose
        private String assertions;

        protected Credential(String type, String token, String authority,
String assertions) {
```

```
            this.type = type;
            this.token = token;
            this.authority = authority;
            this.assertions = assertions;
        }

        public void checkMandatoryFields() {
            if (token == null || token.isEmpty()) {
                throw new InvalidParameterValueException("Resource
CREDENTIAL token field is mandatory");
            }
        }

    }

    private static class Addresses {
        @Expose
        private String url;
        @Expose
        private String name;
        @Expose
        private String port;

        protected Addresses(String url, String name, String port) {
            this.url = url;
            this.name = name;
            this.port = port;
        }
    }

    public static class Host {
        @Expose
        private String id;
        @Expose
        private String address;
        @Expose
        private String agent;
        @Expose
        private String platform;

        protected Host() {
            //
        }

        protected Host(String id, String address, String agent, String
platform) {
            this();
            this.id = id;
            this.address = address;
            this.agent = agent;
            this.platform = platform;
        }
    }

    private static class Geolocation {
        @Expose
        private String id;
        @Expose
        private String latitude;
        @Expose
        private String longtitude;
        @Expose
        private String elevation;
```

```
        @Expose
        private String accurance;
        @Expose
        private String city;
        @Expose
        private String state;
        @Expose
        private String regionICANN;
        @Expose
        private String annotations;

        protected Geolocation(String id, String latittude, String
longtitude, String elevation, String accurance,
                              String city, String state, String
regionICANN, String annotations) {
            this.id = id;
            this.latitude = latittude;
            this.longtitude = longtitude;
            this.elevation = elevation;
            this.accurance = accurance;
            this.city = city;
            this.state = state;
            this.regionICANN = regionICANN;
            this.annotations = annotations;
        }
    }

    private static class Attachements {
        @Expose
        private String contentType;
        @Expose
        private String content;
        @Expose
        private String name;

        protected Attachements(String contentType, String content, String
name) {
            this.contentType = contentType;
            this.content = content;
            this.name = name;
        }

        public void checkMandatoryFields() {
            if ((content == null || content.isEmpty()) ||
                    (contentType == null || contentType.isEmpty()) ) {
                throw new InvalidParameterValueException("Resource
ATTACHEMENTS contentType and content fields " +
                        "are mandatory");
            }
        }

    }

    private static class Endpoint {
        @Expose
        private String url;
        @Expose
        private String name;
        @Expose
        private String port;

        protected Endpoint(String url, String name, String port) {
            this.url = url;
```

```
            this.name = name;
            this.port = port;
        }

        public void checkMandatoryFields() {
            if (url == null || url.isEmpty()) {
                throw new InvalidParameterValueException("Resource ENDPOINT
url field is mandatory");
            }
        }

    }

    public Resource(String typeURI) {
        this.typeURI = typeURI;
        this.credential = new Credential("","","","");
    }

    public Resource(String typeURI, String id) {
        this.typeURI = typeURI;
        this.id = id;
        this.credential = new Credential("","","","");
       }

    public Resource(String typeURI, String id, String userid) {
        this.typeURI = typeURI;
        this.id = id;
        this.userid= userid;
        this.credential = new Credential("","","","");
    }

    public Resource(String typeURI, String id, String name, String host,
Credential credential,
                    String addresses, String userid) {
        this.typeURI = typeURI;
        this.id = id;
        this.name = name;
        //_host = host;
        this.credential = credential;
        this.addresses = addresses;
        this.userid = userid;
    }

    public void checkMandatoryFields() {
        if (id == null || id.isEmpty()) {
            throw new InvalidParameterValueException("Resource " + typeURI
+ " id field is mandatory");
        }
    }

}
```

A.3 Taxonomies.java

```
// Licensed to the Apache Software Foundation (ASF) under one
// or more contributor license agreements.  See the NOTICE file
// distributed with this work for additional information
// regarding copyright ownership.  The ASF licenses this file
// to you under the Apache License, Version 2.0 (the
```

```
// "License"); you may not use this file except in compliance
// with the License.  You may obtain a copy of the License at
//
//   http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied.  See the License for the
// specific language governing permissions and limitations
// under the License.

package com.cloud.event;

import java.util.HashMap;
import java.util.UUID;

public class Taxonomies {
    //eventMapping maps CloudStack Event Category (substring of EventType)
to CADF Target Resource
    public static HashMap<String, String> cstoCadfResourceMapping = new
HashMap<String, String>();
    public static HashMap<Taxonomies.Action, Taxonomies.EventType>
eventActionToTypeMapping = new HashMap<Taxonomies.Action,
            Taxonomies.EventType>();
    public static HashMap<String, String> eventResourcetoUuidMapping = new
HashMap<String, String>();



    public Taxonomies() {

    }

    public enum EventType {
        MONITOR("monitor"),
        ACTIVITY("activity"),
        CONTROL("control");

        private String value;

        EventType(String value) {
            this.value = value;
        }

        public String getValue() {
            return value;
        }
    }

    public enum Action {

        //General Resource Management
        CREATE("create"),
        READ("read"),
        UPDATE("update"),
        DELETE("delete"),

        //Monitoring
        MONITOR("monitor"),

        //Workload and Data Management
        BACKUP("backup"),
```

```
        CAPTURE("capture"),
        CONFIGURE("configure"),
        DEPLOY("deploy"),
        DISABLE("disable"),
        ENABLE("enable"),
        RESTORE("restore"),
        START("start"),
        STOP("stop"),
        UNDEPLOY("undeploy"),

        //Messaging
        RECEIVE("receive"),
        SEND("send"),

        //Security-Identity
        AUTHENTICATE("authenticate"),
        AUTHENTICATE_LOGIN("authenticate/login"),
        RENEW("renew"),
        REVOKE("revoke"),

        //Security, Policy, Access Control
        ALLOW("allow"),
        DENY("deny"),
        EVALUATE("evaluate"),
        NOTIFY("notify"),

        UNKNOWN("unknown");

        private String value;

        Action(String value) {
            this.value = value;
        }

        public String getValue() {
            return value;
        }

    }

    public enum Outcome {
        SUCCESS("success"),
        FAILURE("failure"),
        UNKNOWN("unknown"),
        PENDING("pending");

        private String value;

        Outcome(String value) {
            this.value = value;
        }

        public String getValue() {
            return value;
        }
    }

    public static class Reason {
        public static final String REASON_TYPE = "reasonType";
        public static final String REASON_CODE = "reasonCode";
        public static final String POLICY_TYPE = "policyType";
        public static final String POLICY_CODE = "policyCode";
```

```
    }

    public static class Resource {

        public static final String STORAGE = "storage";
        public static final String STORAGE_NODE = "storage/node";
        public static final String STORAGE_VOLUME = "storage/volume";
        public static final String STORAGE_MEMORY = "storage/memory";
        public static final String STORAGE_MEMORY_CACHE =
"storage/memory/cache";
        public static final String STORAGE_CONTAINER = "storage/container";
        public static final String STORAGE_DIRECTORY = "storage/directory";
        public static final String STORAGE_DATABASE = "storage/database";
        public static final String STORAGE_QUEUE = "storage/queue";

        public static final String COMPUTE = "compute";
        public static final String COMPUTE_NODE = "compute/node";
        public static final String COMPUTE_CPU = "compute/cpu";
        public static final String COMPUTE_CPU_VPU = "compute/cpu/vpu";
        public static final String COMPUTE_MACHINE = "compute/machine";
        public static final String COMPUTE_MACHINE_VM =
"compute/machine/vm";
        public static final String COMPUTE_PROCESS = "compute/process";
        public static final String COMPUTE_THREAD = "compute/thread";

        public static final String NETWORK = "network";
        public static final String NETWORK_NODE = "network/node";
        public static final String NETWORK_NODE_HOST = "network/node/host";
        public static final String NETWORK_NODE_ROUTER =
"network/node/router";
        public static final String NETWORK_NODE_SWITCH =
"network/node/switch";
        public static final String NETWORK_NODE_FIREWALL =
"network/node/firewall";
        public static final String NETWORK_CONNECTION =
"network/connection";
        public static final String NETWORK_CONNECTION_FTP =
"network/connection/ftp";
        public static final String NETWORK_CONNECTION_PIPE =
"network/connection/pipe";
        public static final String NETWORK_DOMAIN = "network/domain";
        public static final String NETWORK_CLUSTER = "network/cluster";

        public static final String DATA = "data";
        public static final String DATA_CATALOG = "data/catalog";
        public static final String DATA_CONFIG = "data/config";
        public static final String DATA_DIRECTORY = "data/directory";
        public static final String DATA_FILE = "data/file";
        public static final String DATA_IMAGE = "data/image";
        public static final String DATA_LOG = "data/log";
        public static final String DATA_MESSAGE = "data/message";
        public static final String DATA_MESSAGE_STREAM =
"data/message/stream";
        public static final String DATA_MODULE = "data/module";
        public static final String DATA_PACKAGE = "data/package";
        public static final String DATA_REPORT = "data/report";
        public static final String DATA_TEMPLATE = "data/template";
        public static final String DATA_WORKLOAD = "data/workload";
        public static final String DATA_WORKLOAD_APPLICATION =
"data/workload/application";
        public static final String DATA_WORKLOAD_SERVICE =
"data/workload/service";
        public static final String DATA_DATABASE = "data/database";
```

```
        public static final String DATA_DATABASE_ALIAS =
"data/database/alias";
        public static final String DATA_DATABASE_INDEX =
"data/database/index";
        public static final String DATA_DATABASE_INSTANCE =
"data/database/instance";
        public static final String DATA_DATABASE_KEY = "data/database/key";
        public static final String DATA_DATABASE_ROUTINE =
"data/database/routine";
        public static final String DATA_DATABASE_SCHEMA =
"data/database/schema";
        public static final String DATA_DATABASE_SEQUENCE =
"data/database/sequence";
        public static final String DATA_DATABASE_TABLE =
"data/database/table";
        public static final String DATA_DATABASE_VIEW =
"data/database/view";
        public static final String DATA_SECURITY = "data/security";
        public static final String DATA_SECURITY_ACCOUNT =
"data/security/account";
        public static final String DATA_SECURITY_ACCOUNT_USER =
"data/security/account/user";
        public static final String DATA_SECURITY_ACCOUNT_ADMIN =
"data/security/account/admin";
        public static final String DATA_SECURITY_CREDENTIAL =
"data/security/credential";
        public static final String DATA_SECURITY_GROUP =
"data/security/group";
        public static final String DATA_SECURITY_IDENTITY =
"data/security/identity";
        public static final String DATA_SECURITY_IDENTITY_ATTRIBUTE =
"data/security/identity/attribute";
        public static final String DATA_SECURITY_IDENTITY_TOKEN =
"data/security/identity/token";
        public static final String DATA_SECURITY_KEY = "data/security/key";
        public static final String DATA_SECURITY_LICENSE =
"data/security/license";
        public static final String DATA_SECURITY_POLICY =
"data/security/policy";
        public static final String DATA_SECURITY_PROFILE =
"data/security/profile";
        public static final String DATA_SECURITY_ROLE =
"data/security/role";
        public static final String DATA_SECURITY_NODE =
"data/security/node";

        public static final String SERVICE = "service";
        public static final String SERVICE_BSS = "service/bss";
        public static final String SERVICE_BSS_BILLING =
"service/bss/billing";
        public static final String SERVICE_BSS_LOCATION =
"service/bss/location";
        public static final String SERVICE_BSS_METERING =
"service/bss/metering";
        public static final String SERVICE_COMPOSITION =
"service/composition";
        public static final String SERVICE_COMPOSITION_ORCHESTRATION =
"service/composition/orchestration";
        public static final String SERVICE_COMPOSITION_WORKFLOW =
"service/composition/workflow";
        public static final String SERVICE_COMPUTE = "service/compute";
        public static final String SERVICE_DATABASE = "service/database";
        public static final String SERVICE_IMAGE = "service/image";
```

```
        public static final String SERVICE_NETWORK = "service/network";
        public static final String SERVICE_OSS = "service/oss";
        public static final String SERVICE_OSS_CAPACITY =
"service/oss/capacity";
        public static final String SERVICE_OSS_CONFIGURATION =
"service/oss/configuration";
        public static final String SERVICE_OSS_LOGGING =
"service/oss/logging";
        public static final String SERVICE_OSS_MONITORING =
"service/oss/monitoring";
        public static final String SERVICE_OSS_PERFORMANCE =
"service/oss/performance";
        public static final String SERVICE_OSS_VIRTUALIZATION =
"service/oss/virtualization";
        public static final String SERVICE_SECURITY = "service/security";
        public static final String SERVICE_STORAGE = "service/storage";
        public static final String SERVICE_STORAGE_BLOCK =
"service/storage/block";
        public static final String SERVICE_STORAGE_OBJECT =
"service/storage/object";

        public static final String SYSTEM = "system";

        public static final String UNKNOWN = "unknown";

    }

    static {
        //Mapping CS Resources (Categories) to CADF Resources

        cstoCadfResourceMapping.put("VM",
Taxonomies.Resource.COMPUTE_MACHINE_VM);
        cstoCadfResourceMapping.put("ROUTER",
Taxonomies.Resource.NETWORK_NODE_ROUTER); //needs check
        cstoCadfResourceMapping.put("PROXY", Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("VNC", Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("NET", Taxonomies.Resource.NETWORK);
        cstoCadfResourceMapping.put("PORTABLE",
Taxonomies.Resource.NETWORK);
        cstoCadfResourceMapping.put("NETWORK",
Taxonomies.Resource.NETWORK);
        cstoCadfResourceMapping.put("FIREWALL",
Taxonomies.Resource.NETWORK);
        cstoCadfResourceMapping.put("FIREWALL.EGRESS",
Taxonomies.Resource.NETWORK);

        cstoCadfResourceMapping.put("NIC",
Taxonomies.Resource.NETWORK_NODE);
        cstoCadfResourceMapping.put("NIC.DETAIL",
Taxonomies.Resource.NETWORK_NODE);

        cstoCadfResourceMapping.put("LB",
Taxonomies.Resource.SERVICE_OSS_PERFORMANCE);
        cstoCadfResourceMapping.put("LB.ASSIGN.TO",
Taxonomies.Resource.SERVICE_OSS_PERFORMANCE);
        cstoCadfResourceMapping.put("LB.REMOVE.FROM",
Taxonomies.Resource.SERVICE_OSS_PERFORMANCE);
        cstoCadfResourceMapping.put("LB.STICKINESSPOLICY",
Taxonomies.Resource.SERVICE_OSS_PERFORMANCE);
        cstoCadfResourceMapping.put("LB.HEALTHCHECKPOLICY",
Taxonomies.Resource.SERVICE_OSS_PERFORMANCE);
        cstoCadfResourceMapping.put("LB.CERT",
Taxonomies.Resource.SERVICE_OSS_PERFORMANCE);
```

```
        cstoCadfResourceMapping.put("GLOBAL.LB",
Taxonomies.Resource.SERVICE_OSS_PERFORMANCE);

        cstoCadfResourceMapping.put("ROLE",
Taxonomies.Resource.DATA_SECURITY_GROUP);
        cstoCadfResourceMapping.put("ROLE.PERMISSION",
Taxonomies.Resource.DATA_SECURITY_GROUP);

        cstoCadfResourceMapping.put("CA.CERTIFICATE",
Taxonomies.Resource.DATA_SECURITY_CREDENTIAL);

        cstoCadfResourceMapping.put("ACCOUNT",
Taxonomies.Resource.DATA_SECURITY_ACCOUNT);
        cstoCadfResourceMapping.put("ACCOUNT.MARK.DEFAULT",
Taxonomies.Resource.DATA_SECURITY_ACCOUNT);

        cstoCadfResourceMapping.put("USER",
Taxonomies.Resource.DATA_SECURITY_ACCOUNT_USER);

        cstoCadfResourceMapping.put("REGISTER.SSH",
Taxonomies.Resource.DATA_SECURITY_KEY);
        cstoCadfResourceMapping.put("REGISTER.USER",
Taxonomies.Resource.DATA_SECURITY_KEY);

        cstoCadfResourceMapping.put("TEMPLATE",
Taxonomies.Resource.DATA_TEMPLATE);
        cstoCadfResourceMapping.put("TEMPLATE.DOWNLOAD",
Taxonomies.Resource.DATA_TEMPLATE);

        cstoCadfResourceMapping.put("VOLUME",
Taxonomies.Resource.STORAGE_VOLUME);
        cstoCadfResourceMapping.put("VOLUME.DETAIL",
Taxonomies.Resource.STORAGE_VOLUME);

        cstoCadfResourceMapping.put("DOMAIN",
Taxonomies.Resource.NETWORK_DOMAIN);

        cstoCadfResourceMapping.put("SNAPSHOT",
Taxonomies.Resource.SERVICE_IMAGE);
        cstoCadfResourceMapping.put("SNAPSHOTPOLICY",
Taxonomies.Resource.SERVICE_IMAGE);

        cstoCadfResourceMapping.put("ISO", Taxonomies.Resource.DATA_IMAGE);

        cstoCadfResourceMapping.put("SSVM", Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("SERVICE.OFFERING",
Taxonomies.Resource.SYSTEM);
        cstoCadfResourceMapping.put("DISK.OFFERING",
Taxonomies.Resource.SYSTEM);
        cstoCadfResourceMapping.put("NETWORK.OFFERING",
Taxonomies.Resource.SYSTEM);

        cstoCadfResourceMapping.put("POD",
Taxonomies.Resource.DATA_SECURITY_GROUP);

        cstoCadfResourceMapping.put("ZONE",
Taxonomies.Resource.DATA_SECURITY_GROUP);

        cstoCadfResourceMapping.put("VLAN.IP.RANGE",
Taxonomies.Resource.SERVICE_NETWORK);
        cstoCadfResourceMapping.put("MANAGEMENT.IP.RANGE",
```

```
Taxonomies.Resource.SERVICE_NETWORK);
        cstoCadfResourceMapping.put("STORAGE.IP.RANGE",
Taxonomies.Resource.SERVICE_NETWORK);

        cstoCadfResourceMapping.put("CONFIGURATION.VALUE",
Taxonomies.Resource.SYSTEM);

        cstoCadfResourceMapping.put("SG",
Taxonomies.Resource.DATA_SECURITY_GROUP);
        cstoCadfResourceMapping.put("SG.AUTH",
Taxonomies.Resource.DATA_SECURITY_GROUP);
        cstoCadfResourceMapping.put("SG.REVOKE",
Taxonomies.Resource.DATA_SECURITY_GROUP);

        cstoCadfResourceMapping.put("HOST",
Taxonomies.Resource.NETWORK_NODE_HOST);
        cstoCadfResourceMapping.put("HOST.OOBM",
Taxonomies.Resource.NETWORK_NODE_HOST);

        cstoCadfResourceMapping.put("HA.RESOURCE",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("HA.STATE",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("MAINT", Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("MAINT.CANCEL",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("MAINT.PREPARE",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("VPN",
Taxonomies.Resource.NETWORK_CLUSTER);
        cstoCadfResourceMapping.put("VPN.REMOTE.ACCESS",
Taxonomies.Resource.NETWORK_CLUSTER);
        cstoCadfResourceMapping.put("VPN.USER",
Taxonomies.Resource.NETWORK_CLUSTER);
        cstoCadfResourceMapping.put("VPN.S2S.VPN.GATEWAY",
Taxonomies.Resource.NETWORK_CLUSTER);
        cstoCadfResourceMapping.put("VPN.S2S.CUSTOMER.GATEWAY",
Taxonomies.Resource.NETWORK_CLUSTER);
        cstoCadfResourceMapping.put("VPN.S2S.CONNECTION",
Taxonomies.Resource.NETWORK_CLUSTER);

        cstoCadfResourceMapping.put("UPLOAD.CUSTOM",
Taxonomies.Resource.DATA_SECURITY_CREDENTIAL);

        cstoCadfResourceMapping.put("STATICNAT",
Taxonomies.Resource.NETWORK_CLUSTER);
        cstoCadfResourceMapping.put("ZONE.VLAN",
Taxonomies.Resource.NETWORK_CLUSTER);

        cstoCadfResourceMapping.put("PROJECT",
Taxonomies.Resource.SERVICE_COMPOSITION);
        cstoCadfResourceMapping.put("PROJECT.ACCOUNT",
Taxonomies.Resource.SERVICE_COMPOSITION);
        cstoCadfResourceMapping.put("PROJECT.INVITATION",
Taxonomies.Resource.SERVICE_COMPOSITION);

        cstoCadfResourceMapping.put("NETWORK.ELEMENT",
Taxonomies.Resource.SERVICE_NETWORK);

        cstoCadfResourceMapping.put("PHYSICAL.NETWORK",
Taxonomies.Resource.NETWORK_DOMAIN);
```

```
        cstoCadfResourceMapping.put("SERVICE.PROVIDER",
Taxonomies.Resource.NETWORK_DOMAIN);

        cstoCadfResourceMapping.put("TRAFFIC.TYPE",
Taxonomies.Resource.NETWORK_DOMAIN);

        cstoCadfResourceMapping.put("PHYSICAL.LOADBALANCER",
Taxonomies.Resource.NETWORK_NODE);

        cstoCadfResourceMapping.put("PHYSICAL.NCC",
Taxonomies.Resource.NETWORK_NODE);

        cstoCadfResourceMapping.put("SWITCH.MGMT",
Taxonomies.Resource.NETWORK_NODE);
        cstoCadfResourceMapping.put("PHYSICAL.FIREWALL",
Taxonomies.Resource.NETWORK_NODE);

        cstoCadfResourceMapping.put("VPC", Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("NETWORK.ACL",
Taxonomies.Resource.DATA_SECURITY_ROLE);

        cstoCadfResourceMapping.put("VPC.OFFERING",
Taxonomies.Resource.SYSTEM);

        cstoCadfResourceMapping.put("PRIVATE.GATEWAY",
Taxonomies.Resource.NETWORK_NODE);

        cstoCadfResourceMapping.put("STATIC.ROUTE",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("CREATE_TAGS",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("DELETE_TAGS",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("CREATE_RESOURCE_DETAILS",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("DELETE_RESOURCE_DETAILS",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("VMSNAPSHOT",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("PHYSICAL",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("PHYSICAL.NVPCONTROLLER",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("PHYSICAL.OVSCONTROLLER",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("PHYSICAL.NUAGE.VSD",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("COUNTER",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("CONDITION",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("AUTOSCALEPOLICY",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("AUTOSCALEVMPROFILE",
```

```
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("AUTOSCALEVMGROUP",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("PHYSICAL.DHCP",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("PHYSICAL.PXE",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("BAREMETAL.RCT",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("BAREMETAL.PROVISION",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("AG", Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("VM.AG", Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("INTERNALLBVM",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("HOST.RESERVATION",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("GUESTVLANRANGE",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("PORTABLE.IP.RANGE",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("PORTABLE.IP",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("DEDICATE",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("DEDICATE.RESOURCE",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("VM.RESERVATION",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("UCS", Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("MIGRATE.PREPARE",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("ALERT", Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("PHYSICAL.ODLCONTROLLER",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("GUEST.OS",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("GUEST.OS.MAPPING",
Taxonomies.Resource.UNKNOWN);

        cstoCadfResourceMapping.put("NIC.SECONDARY.IP",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("EXTERNAL.DHCP.VM.IP",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("USAGE.REMOVE.USAGE",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("NETSCALER.SERVICEPACKAGE",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("NETSCALERVM",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("ANNOTATION",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("TEMPLATE.DIRECT.DOWNLOAD",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("ISO.DIRECT.DOWNLOAD",
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("SYSTEM.VM",
```

```
Taxonomies.Resource.UNKNOWN);
        cstoCadfResourceMapping.put("SYSTEM.MONITOR",
Taxonomies.Resource.DATA_SECURITY);
        cstoCadfResourceMapping.put("EVENT", Resource.DATA_LOG);

        eventActionToTypeMapping.put(Action.CREATE, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.UPDATE, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.DELETE, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.BACKUP, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.CAPTURE, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.CONFIGURE, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.DEPLOY, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.RESTORE, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.START, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.STOP, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.UNDEPLOY, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.RECEIVE, EventType.ACTIVITY);
        eventActionToTypeMapping.put(Action.SEND, EventType.ACTIVITY);

        eventActionToTypeMapping.put(Action.DISABLE, EventType.CONTROL);
        eventActionToTypeMapping.put(Action.ENABLE, EventType.CONTROL);
        eventActionToTypeMapping.put(Action.AUTHENTICATE,
EventType.CONTROL);
        eventActionToTypeMapping.put(Action.AUTHENTICATE_LOGIN,
EventType.CONTROL);
        eventActionToTypeMapping.put(Action.RENEW, EventType.CONTROL);
        eventActionToTypeMapping.put(Action.REVOKE, EventType.CONTROL);
        eventActionToTypeMapping.put(Action.ALLOW, EventType.CONTROL);
        eventActionToTypeMapping.put(Action.DENY, EventType.CONTROL);
        eventActionToTypeMapping.put(Action.EVALUATE, EventType.CONTROL);
        eventActionToTypeMapping.put(Action.NOTIFY, EventType.CONTROL);

        eventActionToTypeMapping.put(Action.MONITOR, EventType.MONITOR);
        eventActionToTypeMapping.put(Action.READ, EventType.MONITOR);
        eventActionToTypeMapping.put(Action.UNKNOWN, EventType.MONITOR);

        //Mapping CS Resources (Categories) to UUIDS
        for (HashMap.Entry he : cstoCadfResourceMapping.entrySet()) {
            eventResourcetoUuidMapping.put(he.getKey().toString(),

UUID.nameUUIDFromBytes(he.getKey().toString().getBytes()).toString());
        }
    }
}
```

A.4 EventTypes.java - Διαφοροποιήσεις

```
package com.cloud.event;

import java.util.HashMap;
import java.util.Map;

import com.cloud.dc.DataCenter;
import com.cloud.dc.Pod;
import com.cloud.dc.StorageNetworkIpRange;
@@ -76,6 +73,9 @@
import org.apache.cloudstack.ha.HAConfig;
import org.apache.cloudstack.usage.Usage;

import java.util.HashMap;
import java.util.Map;

public class EventTypes {

    //map of Event and corresponding entity for which Event is applicable
@@ -90,7 +90,9 @@
    public static final String EVENT_VM_UPDATE = "VM.UPDATE";
    public static final String EVENT_VM_UPGRADE = "VM.UPGRADE";
    public static final String EVENT_VM_DYNAMIC_SCALE = "VM.DYNAMIC.SCALE";
    //TODO change value to VM.PASSWORD.RESET
    public static final String EVENT_VM_RESETPASSWORD = "VM.RESETPASSWORD";
    //TODO change value to VM.SSHKEY.RESET
    public static final String EVENT_VM_RESETSSHKEY = "VM.RESETSSHKEY";
    public static final String EVENT_VM_MIGRATE = "VM.MIGRATE";
    public static final String EVENT_VM_MOVE = "VM.MOVE";
@@ -105,6 +107,7 @@
    public static final String EVENT_ROUTER_REBOOT = "ROUTER.REBOOT";
    public static final String EVENT_ROUTER_HA = "ROUTER.HA";
    public static final String EVENT_ROUTER_UPGRADE = "ROUTER.UPGRADE";
    //TODO change value to ROUTER.DIAGNOSTICS.START
    public static final String EVENT_ROUTER_DIAGNOSTICS =
"ROUTER.DIAGNOSTICS";

    // Console proxy
@@ -113,7 +116,9 @@
    public static final String EVENT_PROXY_START = "PROXY.START";
    public static final String EVENT_PROXY_STOP = "PROXY.STOP";
    public static final String EVENT_PROXY_REBOOT = "PROXY.REBOOT";
    //TODO change value PROXY.HA.UNKNOWN
    public static final String EVENT_PROXY_HA = "PROXY.HA";
    //TODO change value to PROXY.DIAGNOSTICS.START
    public static final String EVENT_PROXY_DIAGNOSTICS =
"PROXY.DIAGNOSTICS";

    // VNC Console Events
@@ -150,7 +155,9 @@
    public static final String EVENT_NIC_DETAIL_REMOVE =
"NIC.DETAIL.REMOVE";

    // Load Balancers
    //TODO change value to LB.RULE.CREATE
    public static final String EVENT_ASSIGN_TO_LOAD_BALANCER_RULE =
"LB.ASSIGN.TO.RULE";
    //TODO change value to LB.RULE.DELETE
    public static final String EVENT_REMOVE_FROM_LOAD_BALANCER_RULE =
"LB.REMOVE.FROM.RULE";
    public static final String EVENT_LOAD_BALANCER_CREATE = "LB.CREATE";
```

```
    public static final String EVENT_LOAD_BALANCER_DELETE = "LB.DELETE";
@@ -192,6 +199,7 @@
    public static final String EVENT_ACCOUNT_CREATE = "ACCOUNT.CREATE";
    public static final String EVENT_ACCOUNT_DELETE = "ACCOUNT.DELETE";
    public static final String EVENT_ACCOUNT_UPDATE = "ACCOUNT.UPDATE";
    //TODO change value to ACCOUNT.ZONE.DEFAULT.UPDATE
    public static final String EVENT_ACCOUNT_MARK_DEFAULT_ZONE =
"ACCOUNT.MARK.DEFAULT.ZONE";

    // UserVO Events
@@ -206,9 +214,11 @@
    public static final String EVENT_USER_LOCK = "USER.LOCK";

    //registering SSH keypair events
    //TODO change value to SSH.KEYPAIR.EVALUATE or SSH.KEYPAIR.ENABLE
    public static final String EVENT_REGISTER_SSH_KEYPAIR =
"REGISTER.SSH.KEYPAIR";

    //register for user API and secret keys
    //TODO change value to SSH.SECRET_API_KEY.EVALUATE or
SSH.SECRET_API_KEY.ENABLE
    public static final String EVENT_REGISTER_FOR_SECRET_API_KEY =
"REGISTER.USER.KEY";

    // Template Events
@@ -244,7 +254,9 @@

    // Snapshots
    public static final String EVENT_SNAPSHOT_CREATE = "SNAPSHOT.CREATE";
    //TODO change value PRIMARY.SNAPSHOT.ENABLE
    public static final String EVENT_SNAPSHOT_ON_PRIMARY =
"SNAPSHOT.ON_PRIMARY";
    //TODO change value PRIMARY.SNAPSHOT.DISABLE
    public static final String EVENT_SNAPSHOT_OFF_PRIMARY =
"SNAPSHOT.OFF_PRIMARY";
    public static final String EVENT_SNAPSHOT_DELETE = "SNAPSHOT.DELETE";
    public static final String EVENT_SNAPSHOT_REVERT = "SNAPSHOT.REVERT";
@@ -267,7 +279,9 @@
    public static final String EVENT_SSVM_START = "SSVM.START";
    public static final String EVENT_SSVM_STOP = "SSVM.STOP";
    public static final String EVENT_SSVM_REBOOT = "SSVM.REBOOT";
    //TODO change value to SSVM.HA.UNKNOWN
    public static final String EVENT_SSVM_HA = "SSVM.HA";
    //TODO change value to SSVM.DIAGNOSTICS.START
    public static final String EVENT_SSVM_DIAGNOSTICS = "SSVM.DIAGNOSTICS";

    // Service Offerings
@@ -342,12 +356,17 @@

    // Maintenance
    public static final String EVENT_MAINTENANCE_CANCEL = "MAINT.CANCEL";
    //TODO change value to MAINTENANCE.PS.STOP
    public static final String EVENT_MAINTENANCE_CANCEL_PRIMARY_STORAGE =
"MAINT.CANCEL.PS";
    //TODO change value to MAINTENANCE.CONFIGURE
    public static final String EVENT_MAINTENANCE_PREPARE = "MAINT.PREPARE";
    //TODO change value to MAINTENANCE.PS.CONFIGURE
    public static final String EVENT_MAINTENANCE_PREPARE_PRIMARY_STORAGE =
"MAINT.PREPARE.PS";

    // Primary storage pool
    //TODO change value to PS.ENABLE
    public static final String EVENT_ENABLE_PRIMARY_STORAGE = "ENABLE.PS";
```

```
    //TODO change value to PS.DISABLE
    public static final String EVENT_DISABLE_PRIMARY_STORAGE =
"DISABLE.PS";

    // VPN
@@ -371,6 +390,7 @@
    public static final String EVENT_NETWORK_RESTART = "NETWORK.RESTART";

    // Custom certificates
    //TODO change value TO CUSTOM.CERTIFICATE.NOTIFY
    public static final String EVENT_UPLOAD_CUSTOM_CERTIFICATE =
"UPLOAD.CUSTOM.CERTIFICATE";

    // OneToOnenat
@@ -458,18 +478,25 @@
    public static final String EVENT_STATIC_ROUTE_DELETE =
"STATIC.ROUTE.DELETE";

    // tag related events
    //TODO change value to TAGS.CREATE
    public static final String EVENT_TAGS_CREATE = "CREATE_TAGS";
    //TODO change value to TAGS.DELETE
    public static final String EVENT_TAGS_DELETE = "DELETE_TAGS";

    // meta data related events
    //TODO change value RESOURCE_DETAILS.CREATE
    public static final String EVENT_RESOURCE_DETAILS_CREATE =
"CREATE_RESOURCE_DETAILS";
    //TODO change value RESOURCE_DETAILS.DELETE
    public static final String EVENT_RESOURCE_DETAILS_DELETE =
"DELETE_RESOURCE_DETAILS";

    // vm snapshot events
    public static final String EVENT_VM_SNAPSHOT_CREATE =
"VMSNAPSHOT.CREATE";
    public static final String EVENT_VM_SNAPSHOT_DELETE =
"VMSNAPSHOT.DELETE";
    //TODO change value to VMSNAPSHOT.PRIMARY.ENABLE
    public static final String EVENT_VM_SNAPSHOT_ON_PRIMARY =
"VMSNAPSHOT.ON_PRIMARY";
    //TODO change value to VMSNAPSHOT.PRIMARY.DISABLE
    public static final String EVENT_VM_SNAPSHOT_OFF_PRIMARY =
"VMSNAPSHOT.OFF_PRIMARY";
    //TODO change value to VMSNAPSHOT.REVOKE or VMSNAPSHOT.UNKNOWN
    public static final String EVENT_VM_SNAPSHOT_REVERT =
"VMSNAPSHOT.REVERTTO";

    // external network device events
@@ -527,14 +554,17 @@
    public static final String EVENT_PORTABLE_IP_TRANSFER =
"PORTABLE.IP.TRANSFER";

    // Dedicated Resources
    //TODO change value to RESOURCE.DEDICATE.CREATE
    public static final String EVENT_DEDICATE_RESOURCE =
"DEDICATE.RESOURCE";
    //TODO change value to RESOURCE.DEDICATE.DELETE
    public static final String EVENT_DEDICATE_RESOURCE_RELEASE =
"DEDICATE.RESOURCE.RELEASE";

    public static final String EVENT_CLEANUP_VM_RESERVATION =
"VM.RESERVATION.CLEANUP";
```

```
    public static final String EVENT_UCS_ASSOCIATED_PROFILE =
"UCS.ASSOCIATEPROFILE";

    // Object store migration
    //TODO change value to SECONDARY.MIGRATE.CONFIGURE
    public static final String EVENT_MIGRATE_PREPARE_SECONDARY_STORAGE =
"MIGRATE.PREPARE.SS";

    //Alert generation
@@ -559,6 +589,7 @@
    public static final String EVENT_NETWORK_EXTERNAL_DHCP_VM_IPFETCH =
"EXTERNAL.DHCP.VM.IP.FETCH";

    //Usage related events
    //TODO change value to USAGE.RECORDS.DELETE
    public static final String EVENT_USAGE_REMOVE_USAGE_RECORDS =
"USAGE.REMOVE.USAGE.RECORDS";

    // Netscaler Service Package events
@@ -577,6 +608,8 @@
    // Diagnostics Events
    public static final String EVENT_SYSTEM_VM_DIAGNOSTICS =
"SYSTEM.VM.DIAGNOSTICS";

    // Event operations Events
    public static final String EVENT_EVENTS_CLEAR = "EVENT.DELETE";
    static {

        // TODO: need a way to force author adding event types to declare
the entity details as well, with out braking
@@ -974,6 +1007,9 @@
        entityEventDetails.put(EVENT_TEMPLATE_DIRECT_DOWNLOAD_FAILURE,
VirtualMachineTemplate.class);
        entityEventDetails.put(EVENT_ISO_DIRECT_DOWNLOAD_FAILURE, "Iso");
        entityEventDetails.put(EVENT_SYSTEM_VM_DIAGNOSTICS,
VirtualMachine.class);

        //Event operations events
        entityEventDetails.put(EVENT_EVENTS_CLEAR, Event.class);
    }

    public static String getEntityForEvent(String eventName) {
```

A.5 ApiServlet.java - Διαφοροποιήσεις

```
package com.cloud.api;

import java.io.UnsupportedEncodingException;
import java.net.InetAddress;
import java.net.URLDecoder;
import java.net.UnknownHostException;
import java.util.Arrays;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.inject.Inject;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.cloud.event.Cadf;
import com.cloud.user.Account;
import com.cloud.user.AccountService;
import com.cloud.user.User;
import com.cloud.utils.HttpUtils;
import com.cloud.utils.StringUtils;
import com.cloud.utils.db.EntityManager;
import com.cloud.utils.net.NetUtils;
import org.apache.cloudstack.api.ApiConstants;
import org.apache.cloudstack.api.ApiServerService;
import org.apache.cloudstack.api.ServerApiException;
@@ -47,14 +36,23 @@
import org.springframework.stereotype.Component;
import org.springframework.web.context.support.SpringBeanAutowiringSupport;

import com.cloud.user.Account;
import com.cloud.user.AccountService;
import com.cloud.user.User;

import com.cloud.utils.HttpUtils;
import com.cloud.utils.StringUtils;
import com.cloud.utils.db.EntityManager;
import com.cloud.utils.net.NetUtils;
import javax.inject.Inject;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.UnsupportedEncodingException;
import java.net.InetAddress;
import java.net.URLDecoder;
import java.net.UnknownHostException;
import java.util.Arrays;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
```

```
import java.util.Map;

@Component("apiServlet")
@SuppressWarnings("serial")
@@ -133,7 +131,9 @@ public void run() {
     }

     void processRequestInContext(final HttpServletRequest req, final
HttpServletResponse resp) {

         InetAddress remoteAddress = null;

         try {
             remoteAddress = getClientAddress(req);
         } catch (UnknownHostException e) {
@@ -154,6 +154,12 @@ void processRequestInContext(final HttpServletRequest
req, final HttpServletResp
         final Map<String, Object[]> params = new HashMap<String,
Object[]>();
         params.putAll(req.getParameterMap());

         Cadf.eventExtraInformation.clear();
         Cadf.eventExtraInformation.put("initiator_host",
remoteAddress.getHostAddress());
         Cadf.eventExtraInformation.put("initiator_method",
req.getMethod());
         Cadf.eventExtraInformation.put("initiator_user-agent",
req.getHeader("user-agent"));
         Cadf.eventExtraInformation.put("initiator_auth-type",
req.getAuthType());

         // For HTTP GET requests, it seems that
HttpServletRequest.getParameterMap() actually tries
         // to unwrap URL encoded content from ISO-9959-1.
         // After failed in using setCharacterEncoding() to control it, end
up with following hacking:
@@ -283,6 +289,8 @@ void processRequestInContext(final HttpServletRequest
req, final HttpServletResp
                     }
                     final User user = entityMgr.findById(User.class,
userId);
                     CallContext.register(user, (Account)accountObj);
                     Cadf.eventExtraInformation.put("initiator_userid",
userId.toString());

Cadf.eventExtraInformation.put("initiator_csAccountName", ((Account)
accountObj).getAccountName());
                 } else {
                     // Invalidate the session to ensure we won't allow a
request across management server
                     // restarts if the userId was serialized to the stored
session
@@ -299,6 +307,8 @@ void processRequestInContext(final HttpServletRequest
req, final HttpServletResp
                 }
             } else {
                 CallContext.register(accountMgr.getSystemUser(),
accountMgr.getSystemAccount());
                 Cadf.eventExtraInformation.put("initiator_userid",
String.valueOf(accountMgr.getSystemUser().getId()));
                 Cadf.eventExtraInformation.put("initiator_csAccountName",
accountMgr.getSystemAccount().getAccountName());
             }
```

```
            if (apiServer.verifyRequest(params, userId, remoteAddress)) {
@@ -308,6 +318,7 @@ void processRequestInContext(final HttpServletRequest
req, final HttpServletResp
                // Add the HTTP method (GET/POST/PUT/DELETE) as well into
the params map.
                params.put("httpmethod", new String[]{req.getMethod()});
                final String response = apiServer.handleRequest(params,
responseType, auditTrailSb);

                HttpUtils.writeHttpResponse(resp, response != null ?
response : "", HttpServletResponse.SC_OK, responseType,
ApiServer.JSONcontentType.value());
            } else {
                if (session != null) {
@@ -322,7 +333,6 @@ void processRequestInContext(final HttpServletRequest
req, final HttpServletResp

apiServer.getSerializedApiError(HttpServletResponse.SC_UNAUTHORIZED,
"unable to verify user credentials and/or request signature", params,
                                responseType);
                HttpUtils.writeHttpResponse(resp, serializedResponse,
HttpServletResponse.SC_UNAUTHORIZED, responseType,
ApiServer.JSONcontentType.value());

            }
        } catch (final ServerApiException se) {
            final String serializedResponseText =
apiServer.getSerializedApiError(se, params, responseType);
@@ -371,4 +381,5 @@ private static String getCorrectIPAddress(String ip) {
        }
        return null;
    }

}
```

A.6 ActionEventUtils.java - Διαφοροποιήσεις

```
package com.cloud.event;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;

import javax.annotation.PostConstruct;
import javax.inject.Inject;

import com.cloud.utils.ReflectUtil;
import com.cloud.utils.db.EntityManager;
import org.apache.cloudstack.api.Identity;
import org.apache.log4j.Logger;
import org.springframework.beans.factory.NoSuchBeanDefinitionException;

import org.apache.cloudstack.context.CallContext;
import org.apache.cloudstack.framework.config.dao.ConfigurationDao;
import org.apache.cloudstack.framework.events.EventBus;
import org.apache.cloudstack.framework.events.EventBusException;

import com.cloud.configuration.Config;
import com.cloud.event.dao.EventDao;
import com.cloud.exception.InvalidParameterValueException;
import com.cloud.projects.Project;
import com.cloud.projects.dao.ProjectDao;
import com.cloud.server.ManagementService;
import com.cloud.user.Account;
import com.cloud.user.AccountVO;
import com.cloud.user.User;
import com.cloud.user.dao.AccountDao;
import com.cloud.user.dao.UserDao;
import com.cloud.projects.dao.ProjectDao;
import com.cloud.projects.Project;
import com.cloud.utils.ReflectUtil;
import com.cloud.utils.component.ComponentContext;
import com.cloud.utils.db.EntityManager;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import org.apache.cloudstack.api.Identity;
import org.apache.cloudstack.context.CallContext;
import org.apache.cloudstack.framework.config.dao.ConfigurationDao;
import org.apache.cloudstack.framework.events.EventBus;
import org.apache.cloudstack.framework.events.EventBusException;
import org.apache.log4j.Logger;
import org.springframework.beans.factory.NoSuchBeanDefinitionException;

import javax.annotation.PostConstruct;
import javax.inject.Inject;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;

public class ActionEventUtils {
    private static final Logger s_logger =
Logger.getLogger(ActionEventUtils.class);
    private static final Logger s_el_logger =
Logger.getLogger("com.cadf.el"); //el stands for event_logger
```

```
    private String _resourceUuid;

    private static EventDao s_eventDao;
    private static AccountDao s_accountDao;
@@ -65,6 +68,7 @@
    public static final String EntityUuid = "entity_uuid";
    public static final String EntityDetails = "entity_details";

    public
    @Inject
    EventDao eventDao;
    @Inject
@@ -183,7 +187,11 @@ private static Event persistActionEvent(Long userId,
Long accountId, Long domain
        if (startEventId != null) {
            event.setStartId(startEventId);
        }

        event = s_eventDao.persist(event);

        createCadfRecord(event);

        return event;
    }

@@ -202,6 +210,7 @@ private static void publishOnEventBus(long userId, long
accountId, String eventC
        // get the entity details for which ActionEvent is generated
        String entityType = null;
        String entityUuid = null;

        CallContext context = CallContext.current();
        //Get entity Class(Example - VirtualMachine.class) from the event
Type eg. - VM.CREATE
        Class<?> entityClass =
EventTypes.getEntityClassForEvent(eventType);
@@ -305,4 +314,29 @@ private static void
populateFirstClassEntities(Map<String, String> eventDescript

    }

    private static void createCadfRecord(EventVO event) {

        /*for (Map.Entry <String, String> entry :
eventExtraInformation.entrySet()) {
            System.out.println("key " + entry.getKey());
            System.out.println("value " + entry.getValue());
        }*/

        Gson gson = new GsonBuilder()
                .excludeFieldsWithoutExposeAnnotation()
                //.setPrettyPrinting()
                .create();

        Cadf cadf = new Cadf(event);


        try {
            cadf.checkMandatoryFields();
        } catch (InvalidParameterValueException e) {
            s_logger.error(e.getMessage());
        }
```

```
        s_el_logger.info(gson.toJson(cadf));

        //System.out.println(gson.toJson(cadf));
    }
}
```

Παράρτημα Β
Πίνακες

Β.1 Αρκτικόλεξα

Αρκτικόλεξο	Ερμηνεία
CADF	Cloud Auditing Data Federation
CFTT	Computer Forensic Tool Testing
CLI	Command Line Interface
CMP	Cloud Management Platform
CMWG	Cloud Management Working Group
CSP	Cloud Service Provider
CnC	Command-n-Control
DDoS	Distributed Denial-of-Service
DFIR	Digital Forensics and Incident Response
DMTF	Distributed Management Task Force
IaaS	Infrastructure as a Service
IoA	Internet of Anything
IoT	Internet of Things
ISACA	Information Systems Audit and Control Association
OMG	Object Management Group
OVF	Open Virtualization Format Working Group
PaaS	Platform as a Service
RDBMS	Relational Database Management System
SaaS	Software as a Service
SDDC	Software Defined Data Center
SEWG	Software Entitlement Working Group
SIEM	Security Information and Event Management

Β.2 Ευρετήριο Σχημάτων

Β.3 Ευρετήριο Στιγμιοτύπων

Β.4. Ευρετήριο Πινάκων

Βιβλιογραφία

[1] J. Dykstra, “Seizing Electronic Evidence from Cloud Computing Environments,” in *Cybercrime and Cloud Forensics: Applications for Investigation Processes*, IGI Global, 2013, pp. 156–185.

[2] M. Iorga and E. Simmon, “DRAFT NISTIR 8006, NIST Cloud Computing Forensic Science Challenges,” National Institute of Standards and Technology, U.S. Department of Commerce, Jun. 2014.

[3] Cloud Auditing Data Federation (CADF) Working Group, “Cloud Auditing Data Federation (CADF) - Data Format and Interface Definitions Specification.” Distributed Management Task Force, Inc. (DMTF), 19-Jun-2014.

[4] H. Alobaidli, Q. Nasir, and M. Abutalib, “CADF Logging Infrastruture For Cloud Computing Digital Forensics,” 2018.

[5] S. Alqahtany, N. Clarke, S. Furnell, and C. Reich, “Cloud Forensics: A Review of Challenges, Solutions and Open Problems,” in *2015 International Conference on Cloud Computing (ICCC)*, 2015, pp. 1–9.

[6] Zafarullah, F. Anwar, and Z. Anwar, “Digital Forensics for Eucalyptus,” in *2011 Frontiers of Information Technology*, 2011, pp. 110–116.

[7] D. Birk and C. Wegener, “Technical Issues of Forensic Investigations in Cloud Computing Environments,” in *2011 Sixth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*, 2011, pp. 1–10.

[8] R. Marty, “Cloud Application Logging for Forensics,” in *Proceedings of the 2011 ACM Symposium on Applied Computing*, New York, NY, USA, 2011, pp. 178–184.

[9] S. Zawoad, R. Hasan, and A. Skjellum, “OCF: An Open Cloud Forensics Model for Reliable Digital Forensics,” in *2015 IEEE 8th International Conference on Cloud Computing*, 2015, pp. 437–444.

[10] H. Lallie and L. Pimlott, “Applying the ACPO Principles in Public Cloud Forensic Investigations,” *Journal of Digital Forensics, Security and Law*, vol. 7, no. 1, Jan. 2012.

[11] M. L. Badger, T. Grance, R. Patt-Corner, and J. Voas, “Cloud computing synopsis and recommendations,” National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-146, 2012.

[12] A. Pătraşcu and V. Valeriu Patriciu, “Logging for Cloud Computing Forensic Systems,” *International Journal of Computers, Communications & Control*, vol. 10, no. 2, pp. 222–229, Apr. 2015.

[13] B. Kumar Raju and G. Geethakumari, “Event correlation in cloud: a forensic perspective,” *Computing*, vol. 98, no. 11, pp. 1203–1224, Nov. 2016.

[14] B. C. Sekhar and G. Murali, “Access control for cloud forensics through secure logging services,” in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017, pp. 3527–3532.

[15] Z. Chen *et al.*, “Secure Logging and Public Audit for Operation Behavior in Cloud Storage,” in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, 2017, vol. 1, pp. 444–450.

[16] H. Alobaidli, Q. Nasir, A. Iqbal, and M. Guimaraes, “Challenges of Cloud Log Forensics,” 2017, pp. 227–230.
[17] M. I. H. Sukmana, K. A. Torkura, F. Cheng, C. Meinel, and H. Graupner, “Unified logging system for monitoring multiple cloud storage providers in cloud storage broker,” in *2018 International Conference on Information Networking (ICOIN)*, 2018, pp. 44–49.
[18] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” National Institute of Standards and Technology, U.S. Department of Commerce, Special Publication 800–145, Sep. 2011.
[19] A. Sears, “5 Benefits of Cloud Technology for Small Businesses in 2018,” 05-Apr-2018. .
[20] K. Panetta, “5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018,” 16-Aug-2018. [Online]. Available: https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/. [Accessed: 22-Dec-2018].
[21] C. Pettey, “Moving to a Software Subscription Model,” 30-May-2018. [Online]. Available: https://www.gartner.com/smarterwithgartner/moving-to-a-software-subscription-model/. [Accessed: 22-Dec-2018].
[22] D. Wall, “Towards a Conceptualisation of Cloud (Cyber) Crime,” 2017, pp. 529–538.
[23] S. Simou, C. Kalloniatis, S. Gritzalis, and H. Mouratidis, “A survey on cloud forensics challenges and solutions,” *Security and Communication Networks*, vol. 9, no. 18, pp. 6285–6314, 2016.
[24] R. Gagliardi, “Security Log Standard - Still an Open Question,” *SCIP Labs*, 15-Mar-2018. .
[25] R. Sturm, C. Pollard, and J. Craig, *Application Performance Management (APM) in the Digital Enterprise: Managing Applications for Cloud, Mobile, IoT and eBusiness*. Morgan Kaufmann, 2017.
[26] R. Gerhards, “The Syslog Protocol,” RFC Editor, RFC5424, Mar. 2009.
[27] “Practical Guide to Cloud Management Platforms.” Cloud Standards Customer Council, 2017.
[28] “Virtualization,” *Wikipedia*. 30-Oct-2018.
[29] “Comparison of platform virtualization software,” *Wikipedia*. 08-Oct-2018.
[30] D. Freet, R. Agrawal, J. J. Walker, and Y. Badr, “Open source cloud management platforms and hypervisor technologies: A review and comparison,” in *SoutheastCon 2016*, 2016, pp. 1–8.
[31] “Hypervisor,” *Wikipedia*. 05-Dec-2018.
[32] S. Naaz and F. Ahmad, “Comparitive Study of Cloud Forensics Tools,” *Communications on Applied Electronics*, vol. 5, no. 3, pp. 24–30, Jun. 2016.
[33] V. Roussev, I. Ahmed, A. Barreto, S. McCulley, and V. Shanmughan, “Cloud forensics–Tool development studies & future outlook,” *Digital Investigation*, vol. 18, pp. 79–95, Sep. 2016.
[34] J. Dykstra and A. T. Sherman, “Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform,” *Digital Investigation*, vol. 10, pp. S87–S95, Aug. 2013.
[35] N. T. Blog, “Netflix SIRT releases Diffy: A Differencing Engine for Digital Forensics in the Cloud,” *Medium*, 17-Jul-2018. .
[36] A. Pătraşcu and V. V. Patriciu, “Logging framework for cloud computing forensic environments,” in *2014 10th International Conference on Communications (COMM)*, 2014, pp. 1–4.

[37] D. K. Konoor, “Auditing in Cloud Computing Solutions with OpenStack,” in *2016 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 2016, pp. 176–176.
[38] R. Basham, G. Chung, M. Rutkowski, and B. Topol, “An Overview of Cloud Auditing Support for OpenStack,” presented at the OpenStack Summit, Atlanta, USA, 13-May-2014.
[39] Cloud Auditing Data Federation (CADF) Working Group, “Cloud Auditing Data Federation - (CADF-OpenStack) - A CADF Representation for OpenStack.” Distributed Management Task Force, Inc. (DMTF), 16-Apr-2015.
[40] S. Bangur and D. Verma, “Adoption of Cloud Auditing Data Federation (CADF) standard by IBM Spectrum Virtualize.” IBM systems, Mar-2017.
[41] D. Smith, L. Leong, and R. Bala, “Magic Quadrant for Cloud Infrastructure as a Service, Worldwide,” Gartner, G00336148, May 2018.
[42] “OpenStack Docs: Middleware Architecture.” [Online]. Available: https://docs.openstack.org/keystonemiddleware/latest/middlewarearchitecture.html. [Accessed: 24-Apr-2019].
[43] “OpenStack Docs: Keystone Event Notifications.” [Online]. Available: https://docs.openstack.org/keystone/latest/admin/event_notifications.html. [Accessed: 21-Jan-2019].
[44] J. Williams, “ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf.” ACPO Crime Business Area.